



João Miguel Reis Morgado

Licenciado em Engenharia Electrotécnica e de Computadores

Sistema de monitorização multisensor para redes de distribuição de água com dispositivos "Pump as a Turbine"

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Orientador: João Pedro Abreu de Oliveira, Professor Auxiliar, FCT-NOVA

Coorientador: João Miguel Murta Pina, Professor Auxiliar, FCT-NOVA

Júri:

Presidente: Doutor João Francisco Alves Martins

Arguente: Doutor Rui Manuel Leitão Santos Tavares

Vogal: Doutor João Pedro Abreu de Oliveira



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

setembro, 2018

Sistema de monitorização multisensor para redes de distribuição de água com dispositivos "Pump as a Turbine"

Copyright © João Miguel Reis Morgado, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

*The powerful play goes on and you can contribute a verse.
In Walt Whitman's "O Me! O Life!"*

Agradecimentos

Gostaria, em primeiro lugar, de agradecer ao meu orientador Professor João Pedro Oliveira e ao meu coorientador João Murta Pina pela disponibilidade demonstrada e auxílio oferecido durante o decorrer do projeto de dissertação.

À Faculdade de Ciências e Tecnologias da Universidade Nova de Lisboa e a todos os professores que nela contribuíram para o meu percurso académico deixo aqui o meu reconhecimento.

Um agradecimento especial aos meus pais sem os quais não teria conseguido fazer este longo percurso. Todo o apoio e esforço que empregaram ajudou e continua a ajudar a formar-me enquanto pessoa e por isso estarei eternamente grato.

A todos os meus amigos que me ajudaram a subir esta montanha, que foram um apoio constante em todos os momentos, um sincero obrigado. À Ana Domingos, Diana Marisa Rodrigues e Sara Felgueiras, amigas de longa data que me conhecem melhor do que eu mesmo. Um agradecimento especial aos meus irmãos de faculdade Alexandre Dias, Beatriz Salvado, Bruno Santos, Carolina Lagartinho de Oliveira, David Alexandre e Pedro Guerreiro por todas as batalhas académicas em que entrámos juntos.

Por último quero agradecer à minha namorada pelo apoio e paciência ilimitados dando-me sempre a confiança necessária para seguir em frente.

Resumo

A extração de energia elétrica a partir de energia hídrica está, habitualmente, associada a centrais hidroelétricas de grande capacidade. Todavia, a expansão deste ecossistema hidroelétrico terá de incluir de forma mais consistente, micro-geração elétrica a partir de pequenas represas, cursos de água de fluxo reduzido e redes públicas de distribuição de água. A instalação deste tipo de sistemas de micro-geração muitas vezes, colide com as dificuldades em obter microturbinas adequadas à produção pretendida. Nesse sentido, foram sendo adaptadas bombas de água no seu funcionamento inverso. A inexistência de características definidoras das bombas de água no seu funcionamento inverso e a imprecisão dos métodos de predição existentes, tornam a monitorização em tempo real a melhor possibilidade de desenvolver e controlar estes sistemas.

O contínuo desenvolvimento de sensores e tecnologias de comunicação e processamento trouxeram uma nova possibilidade para a criação de sistemas de monitorização de baixo custo. Neste projeto apresenta-se o desenvolvimento, implementação e teste de um nó de monitorização multisensor e adaptável. O nó recorre a circuitos integrados de última geração e reconfiguráveis seguindo uma arquitetura do tipo System-On-Chip (SoC) com ganhos ao nível da ocupação de espaço, flexibilidade e custo.

Palavras-chave: Sistemas de distribuição de água, Bomba em funcionamento de turbina, Internet das coisas, Monitorização

Abstract

Extraction of electrical energy from hydropower is usually associated with large-capacity hydroelectric plants. However, the expansion of this hydroelectric ecosystem will have to include, more consistently, small size electrical generators compatible with small dams, low-flow rate rivers and public water distribution networks. The installation of this type of micro-generation systems often conflicts with the difficulties in obtaining microturbines suitable for the intended production. In this sense, water pumps have been adapted in their reverse operation. The lack of defining characteristics of the water pumps in their reverse operation and the imprecision of existing prediction methods make real-time monitoring the best possibility to develop and control these systems.

The continuous development of sensors and communication and processing technologies has brought a new possibility for the creation of low-cost monitoring systems. This project presents the development, implementation and testing of a multisensor and adaptive monitoring node. The node uses state-of-the-art and reconfigurable integrated circuits following a System-On-Chip (SoC) architecture with gains in space occupancy, flexibility and cost.

Keywords: Water Distribution Systems, Pump as Turbine (PaT), Internet of Things, Monitoring

Índice

AGRADECIMENTOS.....	VII
SIGLAS.....	XVI
1. INTRODUÇÃO.....	1
1.1. MOTIVAÇÃO.....	1
1.2. OBJETIVOS E CONTRIBUIÇÕES.....	2
1.3. ESTRUTURA DO DOCUMENTO	3
2. FUNDAMENTOS TECNOLÓGICOS.....	5
2.1. MICRO-GERAÇÃO HÍDRICA.....	5
2.1.1. <i>Classificação de Turbinas</i>	6
2.1.2. <i>Comparação turbinas/PaT</i>	7
2.1.3. <i>Caracterização de micro-geradores do tipo PaT</i>	9
2.2. A INTERNET DAS COISAS (IOT) NA MONITORIZAÇÃO DE SISTEMAS	11
2.2.1. <i>Microcontrolador (Microcontroller Unit -MCU)</i>	12
2.2.1.1. Arduino.....	13
2.2.1.2. Raspberry Pi.....	13
2.2.1.3. PSoC.....	14
2.2.1.4. ESP8266	15
2.2.1.5. <i>Análise comparativa</i>	16
2.2.2. <i>Módulo de gestão de potência (Power Management Unit – PMU)</i>	17
2.2.2.1 LDO	17
2.2.2.2. Conversores DC/DC.....	18
2.2.2.3. Conversores de Nível Lógico.....	18
2.2.3. <i>ADC (Analog-Digital Converter)</i>	19
2.2.3.1. SAR (Successive Approximation Register) ADC.....	19
2.2.3.2. ADC Sigma-Delta.....	20
2.2.4. <i>Módulo de Comunicação</i>	21
2.2.4.1. Protocolos Locais.....	22
2.2.4.1.1. <i>Universal Asynchronos Receiver and Transmitter (UART)</i>	22
2.2.4.1.2. <i>Serial Peripheral Interface (SPI)</i>	22
2.2.4.1.3. <i>Inter-Integrated Circuit (I2C)</i>	23
2.2.4.2. Protocolos de Rede.....	24
2.2.4.2.1. <i>IEEE 802.15.1 (Bluetooth)</i>	24

2.2.4.2.2. IEEE 802.11 (Wi-Fi)	25
2.2.4.2.3. IEEE 802.15.4 (ZigBee)	26
2.2.5. Módulo de Sensores.....	27
2.2.5.1. Sensor de Caudal	27
2.2.5.2. Sensor de Pressão	29
2.2.5.3. Sensor de Temperatura	29
3. ARQUITETURA E IMPLEMENTAÇÃO DO SISTEMA	31
3.1. NÓ DE AGREGAÇÃO	31
3.1.1. Medição de temperatura.....	32
3.1.2. Medição de pressão	34
3.1.3. Medição do caudal.....	35
3.1.5. Envio da informação	38
3.1.5. Disposição dos pinos	38
3.2. NÓ DE COMUNICAÇÃO.....	39
3.3. CONVERSOR DC-DC	41
3.4. CONVERSOR DE NÍVEL LÓGICO	41
3.5. ELABORAÇÃO E IMPLEMENTAÇÃO DO PROTÓTIPO.....	42
3.6. BASE DE DADOS E SERVIDOR	45
3.7. INTERFACE GRÁFICA	48
4. RESULTADOS EXPERIMENTAIS.....	49
4.1. BYPASS	50
4.2. PAT	54
5. CONCLUSÕES E TRABALHOS FUTUROS.....	59
5.1. SÍNTESE.....	59
5.2. TRABALHOS FUTUROS	60
BIBLIOGRAFIA.....	61
CÓDIGO PSOC.....	65
CÓDIGO ESP8266.....	67
CÓDIGO SERVIDOR.....	71

Lista de Figuras

FIGURA 1.1- SISTEMA A MONITORIZAR	3
FIGURA 2.1- TURBINA DE IMPULSO (ESQUERDA) E REAÇÃO (DIREITA). ADAPTADA DE [2]	6
FIGURA 2.2- COMPARAÇÃO DE PERFORMANCE ENTRE TURBINAS E PAT[2]	8
FIGURA 2.3- COMPARAÇÃO DA PERFORMANCE DE BOMBAS DE ÁGUA EM DIFERENTES MODOS DE FUNCIONAMENTO[3].....	9
FIGURA 2.4-ARQUITETURA DE NÓ IoT	12
FIGURA 2.5- PLACAS ARDUINO. ADAPTADO DE [6].....	13
FIGURA 2.6- RASPBERRY PI MODELO 3 [7].....	14
FIGURA 2.7- PSOC 5LP [10].....	15
FIGURA 2.8- ESP8266 ESP-12E [12].....	15
FIGURA 2.9- ARQUITETURA TÍPICA LDO.....	17
FIGURA 2.10- ESQUEMÁTICO CONVERSOR DE NÍVEL LÓGICO [13].....	18
FIGURA 2.11- ALGORITMO DE COMPARAÇÃO SAR. ADAPTADO DE APONTAMENTOS.....	20
FIGURA 2.12- EFEITOS DA SOBRE AMOSTRAGEM NO ESPETRO DA FREQUÊNCIA. ADAPTADO DE [16]	21
FIGURA 2.13- NOISE SHAPING [16]	21
FIGURA 2.14- FORMATO DE <i>FRAME</i> UART [17].....	22
FIGURA 2.15- TOPOLOGIA SPI[18]	23
FIGURA 2.16- TOPOLOGIA I2C [19].....	24
FIGURA 2.17-TOPOLOGIAS <i>BLUETOOTH</i> [21].....	25
FIGURA 2.18- REDE SEM FIOS COM <i>ACCESS POINT</i> [23].....	26
FIGURA 2.19- TOPOLOGIAS <i>ZIGBEE</i> [25]	26
FIGURA 2.20- SÉRIE FPR-200 [27].....	28
FIGURA 2.21- SÉRIE FTB8000B [28].....	28
FIGURA 2.22- FUNCIONAMENTO BÁSICO DO MEDIDOR. ADAPTADO DE [29]	28
FIGURA 2.23- SÉRIE PXM209 [30]	29
FIGURA 2.24- SEN0257 [31]	29
FIGURA 2.25- DS18B20 [32].....	30
FIGURA 2.26- MNS-9-W1-TS-WT-L03 [33]	30
FIGURA 3.1- LIGAÇÃO ENTRE DS18B20 E MICROCONTROLADOR [37].....	32
FIGURA 3.2- CIRCUITO DE OBTENÇÃO DE VALORES DE TEMPERATURA	33
FIGURA 3.3- ESTADOS DO BLOCO DS18x8 [34]	33
FIGURA 3.4- CARACTERÍSTICA SEN0257. ADAPTADO DE [35]	34

FIGURA 3.5- CIRCUITO DE OBTENÇÃO DOS VALORES DE PRESSÃO.....	34
FIGURA 3.6- FLUXOGRAMA DO ALGORITMO DE MEDIÇÃO IMPLEMENTADO	35
FIGURA 3.7- CARACTERÍSTICA YF-DN50. ADAPTADO DE [36].....	36
FIGURA 3.8- CIRCUITO DE OBTENÇÃO DO VALOR DO CAUDAL.....	37
FIGURA 3.9- JANELA GERADA PELO BLOCO <i>PWM</i>	37
FIGURA 3.10- MÓDULO UART	38
FIGURA 3.11- DISTRIBUIÇÃO DOS PINOS NA PLACA PSoC. ADAPTADO DE [38]	39
FIGURA 3.12- DISTRIBUIÇÃO DE PINS NA PLACA ESP8266. ADAPTADO DE [39].....	40
FIGURA 3.13- CONVERSOR DE NÍVEL LÓGICO BIDIRECIONAL. ADAPTADO DE [37].....	42
FIGURA 3.14- ESQUEMÁTICO DA IMPLEMENTAÇÃO.....	43
FIGURA 3.15- PROTÓTIPO UTILIZADO NO SISTEMA.....	43
FIGURA 3.16- MONTAGEM SENSOR DE PRESSÃO: 1. SENSOR, 2. ANEL ADAPTADOR DN50.....	44
FIGURA 3.17- MONTAGEM DO SENSOR DE CAUDAL: 1. SENSOR, 2. ADAPTADOR M50 2'', 3. ADAPTADOR MACHO.....	44
FIGURA 3.18- ESQUEMA RELACIONAL DA BASE DE DADOS.....	46
FIGURA 3.19- FLUXOGRAMA DO ALGORITMO DE ARMAZENAMENTO IMPLEMENTADO.....	47
FIGURA 3.20- INTERFACE MATLAB.....	48
FIGURA 4.1- ESQUEMA DETALHADO DA MONTAGEM.....	49
FIGURA 4.2- AUTOTRANSFORMADOR TRIFÁSICO	50
FIGURA 4.3- MEDIÇÃO DO FLUXO EM MODO BYPASS	51
FIGURA 4.4- MEDIÇÃO DAS TEMPERATURAS EM MODO BYPASS	53
FIGURA 4.5- VARIAÇÃO DAS PRESSÕES EM MODO <i>BYPASS</i>	53
FIGURA 4.6- VARIAÇÃO DAS PRESSÕES EM MODO <i>PAT</i>	55
FIGURA 4.7- MANÓMETRO DE ENTRADA (ESQUERDA) E SAÍDA (DIREITA) NO PONTO 4.....	57

Lista de Tabelas

TABELA 2.1- CLASSIFICAÇÃO DE DIFERENTES TIPOS DE TURBINAS. ADAPTADO DE [3].....	7
TABELA 2.2- PREDIÇÃO DAS CARACTERÍSTICAS DE PAT. ADAPTADO DE [4]	10
TABELA 2.3- COMPARAÇÃO DE MCUS.....	16
TABELA 2.4- SENSORES DE CAUDAL.....	28
TABELA 2.5- SENSORES DE PRESSÃO.....	29
TABELA 2.6- SENSORES DE TEMPERATURA	30
TABELA 4.1- REGISTO DAS MEDIÇÕES DE FLUXO (L/MIN) EFETUADAS.....	52
TABELA 4.2-REGISTO DAS MEDIÇÕES DE PRESSÃO (KPA) EFETUADAS.....	56

Siglas

Sigla	Descrição
PaT	<i>Pump as Turbine</i>
IoT	<i>Internet of Things</i>
ERT	<i>Energy Recovery Turbine</i>
SHP	<i>Small HydroPower</i>
CFD	<i>Computacional Fluid Dynamics</i>
SoC	<i>System- on- Chip</i>
MCU	Micro Controler Unit
BEP	<i>Best Efficiency Point</i>



Introdução

1.1.Motivação

Com uma capacidade instalada¹ superior a 1 TW a nível mundial, a geração de energia elétrica a partir da energia hídrica representou, por exemplo, em 2016 mais de 70% de toda a energia elétrica obtida a partir de fontes renováveis. Mesmo quando comparada com outras fontes, as centrais hidroelétricas produziram mais de 16% da toda energia elétrica produzida por fontes renováveis e não-renováveis.

A extração de energia elétrica a partir de energia hídrica está, habitualmente, associada a centrais hidroelétricas de grande capacidade. Todavia, a expansão deste ecossistema hidroelétrico terá de incluir de forma mais consistente, micro-geração elétrica a partir de pequenas represas, cursos de água de fluxo reduzido e redes públicas de distribuição de água.

¹ <https://www.worldenergy.org/data/resources/resource/hydropower/>

O nível de penetração deste tipo de solução no mercado das micro centrais de produção elétrica está fortemente dependente da facilidade de instalação, baixo custo e manutenção reduzida. Tomando como referência as centrais de grande capacidade, um dos elementos do sistema com maior relevância é turbina e respetivo gerador elétrico que lhe está acoplado. Todavia, este tipo equipamento, dada sua natureza, apresenta custos tendencialmente não adequados para o efeito comercial pretendido.

Em alternativa às turbinas convencionais, renovou-se, mais recentemente, um novo interesse científico e tecnológico sobre funcionamento de bombas de água normais em modo inverso (turbina), *pump as turbine* (PaT). Estas bombas normais já existentes no mercado (com custos inferiores) evitam a utilização de geradores com turbinas dedicadas e de elevado custo nos projetos de desenvolvimentos de solução de micro/mini-hídricas.

A dificuldade de construção de modelos adequados sobre o comportamento dinâmico das PaT, a operarem perto do seu ponto de funcionamento ótimo, suscita a necessidade acoplar um microssistema de monitorização flexível e de baixo custo capaz de determinar qual o seu ponto ótimo de funcionamento e adaptar as variáveis de controlo do sistema.

Por outro lado, a existência de vários focos de produção independentes requer, de maneira a melhor responder às necessidades de consumo em tempo real e de vigilância sobre o próprio equipamento, um sistema de monitorização autónomo capaz de centralizar a informação proveniente dos vários sistemas ativos. Através de uma monitorização adequada, é espectável que os próprios sistemas possam ser melhor aproveitados aumentando a sua durabilidade e eficiência.

O advento tecnológico permitiu uma maior facilidade na integração de sistemas de aquisição de dados com módulos de comunicação e armazenamento de dados. Este avanço permitiu uma aposta mais firme no desenvolvimento e aplicação de sistemas IoT (*Internet of Things*).

1.2. Objetivos e contribuições

O projeto, objeto desta dissertação, inclui a criação e implementação de um sistema multisensor capaz de monitorizar o comportamento de soluções de micro-geração hídrica suportada em dispositivos PaT.

O sistema de laboratório onde o protótipo é instalado depois de desenvolvido e instalado é, tal como demonstrado na Figura 1.1, constituído por uma bomba de água e um circuito fechado, alimentados por um reservatório.

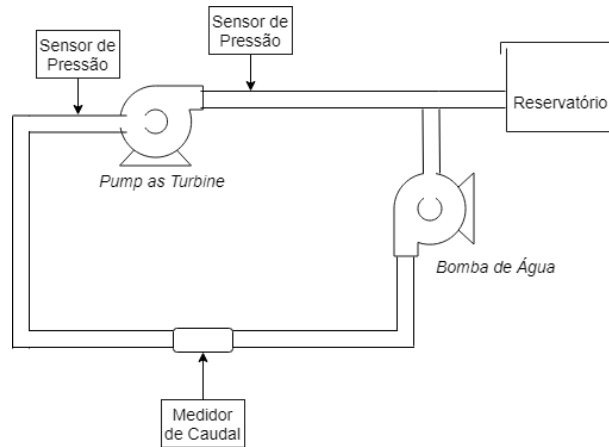


Figura 1.1- Sistema a monitorizar

Devido à importância de variáveis como o caudal e a pressão na segurança do sistema e na produção de energia estas serão as principais medições a serem recolhidas.

A principal contribuição deste projeto é a implementação de um nó de monitorização flexível e de fácil instalação.

1.3.Estrutura do documento

A estrutura da presente dissertação divide-se em 5 capítulos, sendo este o que introduz o tema e o respetivo enquadramento e motivação.

No capítulo 2, são abordados um conjunto de fundamentos tecnológicos sobre a micro-geração recorrendo a motores de água a funcionamento em modo inverso.

No capítulo 3 são apresentados o dimensionamento e a implementação do nó de monitorização

No capítulo 4 são apresentados os resultados experimentais obtidos com o sistema completo.

Finalmente no capítulo 5 são traçadas um conjunto de conclusões e enumeradas alguns de eixos de desenvolvimentos futuros.



Fundamentos Tecnológicos

Neste capítulo far-se-á uma revisão dos aspetos fundamentais da conversão energética através de sistemas hídricos e, mais detalhadamente, através de sistemas PaT.

2.1. Micro-geração hídrica

A energia hídrica compõe, de acordo com o *World Energy Council*, 71% da energia renovável global. A conversão energética pode ser obtida tanto a partir da energia potencial de quedas de água presentes em barragens como da energia cinética proveniente de diversos fluxos de água. Grande parte da produção hídrica encontra-se atualmente centralizada em centrais de grande produção não possibilitando o acesso a zonas e populações onde a rede de distribuição ainda não está desenvolvida ou sequer implementada. Como maneira de potenciar uma solução viável e sustentável para esta problemática, têm sido desenvolvidos estudos e projetos de baixa produção hídrica (*small hydropower*, *SHP*).

As variações causadas pela passagem da água (ou outro fluido) através de qualquer elemento capaz de modificar o seu comportamento, podem ser descritas através da equação de Bernoulli:

$$Z_x + \frac{p_x}{\gamma} + \frac{v_x^2}{2g} = Z_y + \frac{p_y}{\gamma} + \frac{v_y^2}{2g} + \Delta H \quad (2.1)$$

Onde Z (m) corresponde à altura do ponto em relação ao plano de referência, p (Pa) à pressão do fluxo no ponto, γ (N/m^3) ao peso específico do fluido, v (m/s) à velocidade do fluido, g (m/s^2) à aceleração gravítica e ΔH à variação de energia entre os pontos.

As transferências da energia de e para o fluido dependem das máquinas utilizadas no sistema hídrico. Globalmente podem ser classificadas como bombas de água as máquinas que introduzem energia no sistema e turbinas as que retiram energia do sistema.

2.1.1. Classificação de Turbinas

A classificação de turbinas pode ser dividida em dois grandes grupos, turbinas de impulso e de reação. As turbinas de impulso utilizam um fluxo tangencial para criar rotação no rotor da máquina. A conversão da energia cinética presente no fluido para energia mecânica no rotor torna este tipo de turbinas indicadas para casos de elevada queda de água.[1]

Nas turbinas de reação, as pás encontram-se imersas no fluxo sendo que apenas uma pequena parte da queda de pressão se dá anteriormente à entrada na turbina. As válvulas de libertação do fluido estão conectadas ao rotor; dessa maneira o fluido, ao ser liberto, produz uma força de reação que impele o movimento do rotor.

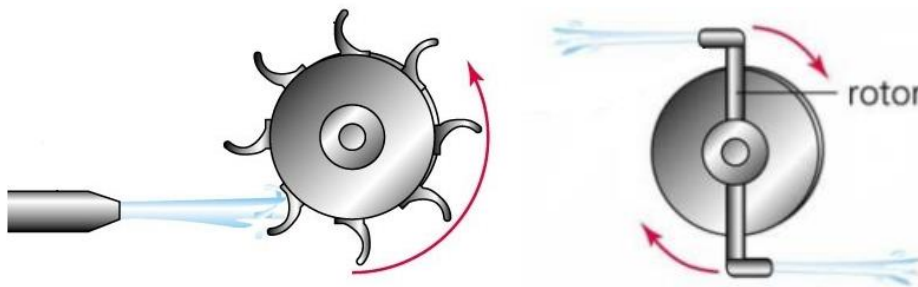


Figura 2.1- Turbina de impulso (esquerda) e reação (direita). Adaptada de [2]

A caracterização apresentada na Tabela 2.1 faz uma distribuição de algumas das turbinas existentes, organizando-as tendo pelos dois grupos previamente explicitados como pela altura da queda de água para o qual é indicado o seu uso.

Tabela 2.1- Classificação de diferentes tipos de turbinas. Adaptado de [3]

Tipo de Turbina	Classificação da Queda		
	Alta (>50m)	Média (10-50m)	Baixa (<10m)
Impulso	Pelton Turgo	<i>Crossflow</i> Turgo Pelton	<i>Crossflow</i>
Reação		Francis <i>Pump-as-Turbine</i>	<i>Propeller</i> Kaplan Francis

2.1.2. Comparação turbinas/PaT

Como alternativa às turbinas convencionais tem sido estudado o funcionamento de bombas de água em modo inverso (turbina), *pump as turbine*. A adaptação de bombas de água tradicionais permite ampliar a gama de pressões e quedas de água nos quais os sistemas de aproveitamento hídrico podem ser utilizados. Este desenvolvimento permitirá uma evolução em centrais SHP, onde a sua principal limitação é a indisponibilidade de microturbinas de baixa capacidade.

Globalmente falando é possível perceber, a partir da Figura 2.2, as diferenças de rendimento entre a utilização de turbinas convencionais e bombas em funcionamento de turbina. As turbinas convencionais permitem a obtenção de maiores rendimentos para uma vasta gama de valores de caudal da turbina (Q_T) em relação ao seu caudal ótimo (Q_{opt}) sendo que a PaT apenas atinge rendimentos da mesma ordem de grandeza com relações $Q_T/Q_{T,OPT}$ entre os 80-100%.

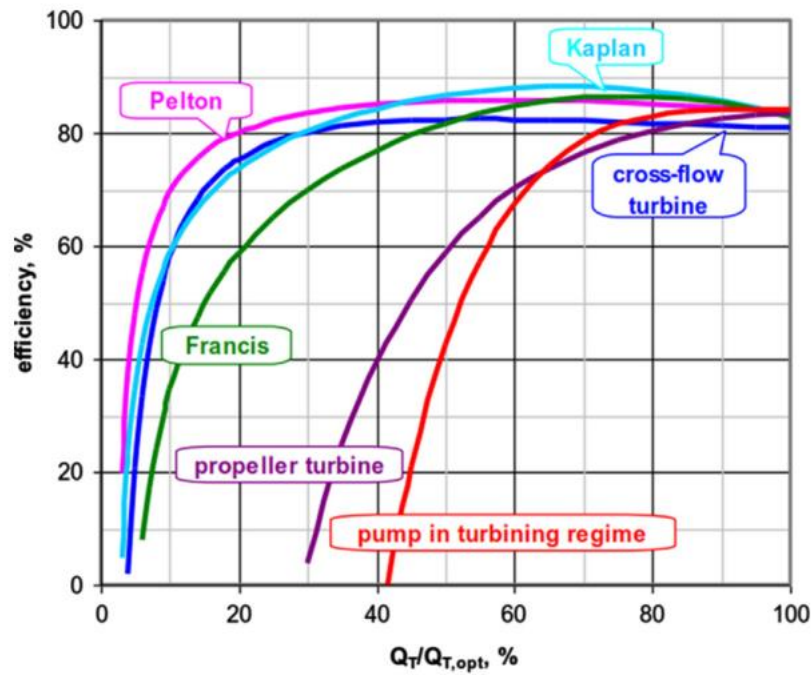


Figura 2.2- Comparação de performance entre turbinas e PaT[2]

Visto as bombas em regime turbina apenas apresentarem rendimentos consistentes com os das restantes turbinas para uma curta gama de valores $Q_T/Q_{T,opt}$, o seu uso é particularmente indicado para aplicações de carga fixa, perto dos valores máximos comportados pela bomba. [2]

Apesar das diferentes características de rendimento entre turbinas em modo convencional e bombas em modo inverso, o estudo [3] demonstra que os rendimentos são quase idênticos quando comparados os funcionamentos da bomba no modo direto e inverso. Os resultados são visíveis na Figura 2.3 onde o rendimento (η) surge em função do *flow rate number* (ϕ), grandeza calculada segundo a equação 2.2.

$$\phi = \frac{Q}{nD^3} \quad (2.2)$$

Onde Q é o fluxo (m^3/s), n a velocidade de rotação (rps) e D o diâmetro do rotor (m).

A experiência[3] foi realizada sobre várias PaT centrífugas representadas no gráfico pelos seus valores de velocidade específica (N_s). A velocidade específica é uma grandeza adimensional, teórica, utilizada para caracterizar turbomaquinaria.

$$N_s = \frac{n\sqrt{Q}}{H^{3/4}} \quad (2.3)$$

Onde Q é o fluxo (m^3/s), n a velocidade de rotação (rps) e H a queda útil (m).

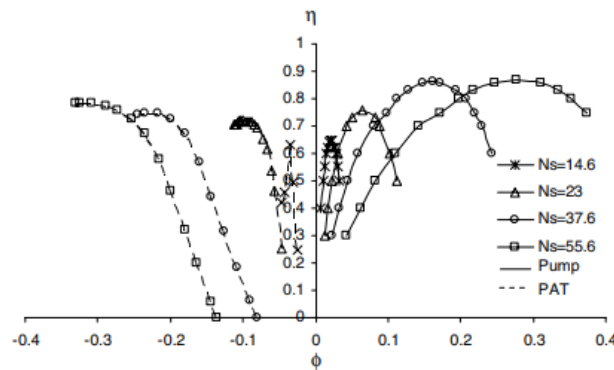


Figura 2.3- Comparação da performance de bombas de água em diferentes modos de funcionamento[3]

Não obstante as limitações de rendimento acima apresentadas pelas PaT em relação a turbinas convencionais estas apresentam também vantagens devido aos reduzidos custos, menor complexidade de funcionamento e maior variedade disponível em termos de características.

Tendo em conta todos os fatores, as PaT são apresentadas como uma solução economicamente viável para cenários de baixa produção como zonas rurais/remotas ou aplicadas a infraestruturas urbanas já existentes de redes de distribuição de água.

2.1.3. Caracterização de micro-geradores do tipo PaT

O principal obstáculo aquando a projeção e dimensionamento de um sistema com utilização de uma PaT é a seleção da bomba de água a ser utilizada.

A falta de curvas que caracterizem as diferentes bombas de água no seu modo inverso (turbina) motivou a realização de vários testes, experimentais e teóricos, de maneira a melhor prever as características de funcionamento e performance das PaT.

Foi realizada, por [4], uma análise de vários métodos de predição baseados nos critérios de velocidade específica da bomba (N_S) e ponto de maior eficiência (BEP). Dos métodos estudados, resumidos na Tabela 2.2, nenhum conseguiu caracterizar o funcionamento das PaT para as suas gamas de velocidades específicas sendo que apenas os métodos apresentados por Stepanoff e Sharma apresentaram resultados concordantes com os dados experimentais para N_S compreendidos entre os 40 e os 60.

Estudos mais recentes recorreram à tecnologia CFD (*Computational Fluid Dynamics*) de maneira a obter modelos funcionais para o funcionamento das PaT. O método de predição utilizado em [5], apesar de obter resultados com uma margem de erro entre os 1,1 e os 5,25% para o funcionamento em modo direto (bomba), obteve resultados não coincidentes com os dados experimentais para o modo inverso (turbina), apresentando um erro máximo de 22,9%.

Tabela 2.2- Predição das características de PaT. Adaptado de [4]

Investigador	Critério	Relação de Queda (H_t/H_p)	Relação de Caudal (Q_t/Q_p)
Stepanoff	BEP	$\frac{1}{\eta_p}$	$\frac{1}{\sqrt{\eta_p}}$
Alatorre-Frank	BEP	$\frac{1}{0,85\eta_p^5 + 0,358}$	$\frac{0,85\eta_p^5 + 0,358}{2\eta_p^{9,5} + 0,205}$
Schmiedl	BEP	$-1,4 + \frac{2,5}{\eta_{hp}}$	$-1,5 + \frac{2,4}{\eta_{hp}^2}$
Grover	N_S	$2,693 - 0,0229N_{st}$	$2397 - 0,0264N_{st}$
Sharma	BEP	$\frac{1}{\eta_p^{1,2}}$	$\frac{1}{\eta_p^{0,8}}$
Hergt	N_S	$1,3 - \frac{6}{N_{st} - 3}$	$1,3 - \frac{1,6}{N_{st} - 5}$
Childs	BEP	$\frac{1}{\eta_p}$	$\frac{1}{\eta_p}$
Hancock	BEP	$\frac{1}{\eta_t}$	$\frac{1}{\eta_t}$

A falta de modelos adequados de previsão e a necessidade das PaT operarem perto do seu BEP, leva estes sistemas a necessitarem de uma monitorização capaz de perceber qual o seu ponto ótimo de funcionamento e adaptar as variáveis de controlo do sistema a esses dados.

O progressivo desenvolvimento de redes de sensores em conjunto com tecnologias de comunicação e processamento de baixo custo, impulsionou o controlo de sistemas físicos através de estruturas *IoT*.

2.2. A Internet das Coisas (IoT) na monitorização de sistemas

As tecnologias IoT (*Internet of Things*), definidas pela ITU (*International Telecommunication Union*) como uma infraestrutura global capaz de providenciar serviços avançados de interconexão têm sido aplicadas, nos últimos anos, nas mais diversas áreas de utilização. Isto deve-se a uma relação cada vez mais proeminente entre a sociedade e a tecnologia.

As áreas mais relevantes de aplicação incluem atualmente:

- **Industria (Industria 4.0);**

Promove a execução e otimização, através de sistemas de produção inteligentes, de “fábricas inteligentes”. Os sistemas físicos de produção, fundidos com sistemas eletrónicos, são capazes de monitorizar e regular as necessidades de manufatura.

- **Domicílios Privados;**

Proporciona ao utilizador a possibilidade de verificar e controlar todo e qualquer dispositivo ligado à rede, bem como monitorizar a própria área física domiciliária.

- **Cidades;**

São semelhantes aos sistemas aplicados nos domicílios privados, mas em maior escala. Cada espaço urbano pode ser gerido em relação aos seus serviços utilitários, nomeadamente fornecimento de águas, eletricidade e redes de iluminação.

- **Sistemas de Saúde.**

Os sistemas desenvolvidos nesta área divergem desde a monitorização de doenças crónicas (períodos mais alargados) e sinais vitais (questões de diagnóstico ou acompanhamento) ao controlo da dosagem ingerida, através do chamado “comprimido digital.

Neste capítulo serão descritos os blocos base que compõem as tecnologias IoT (Figura 2.4) bem como as normas mais utilizados em cada um desses módulos.

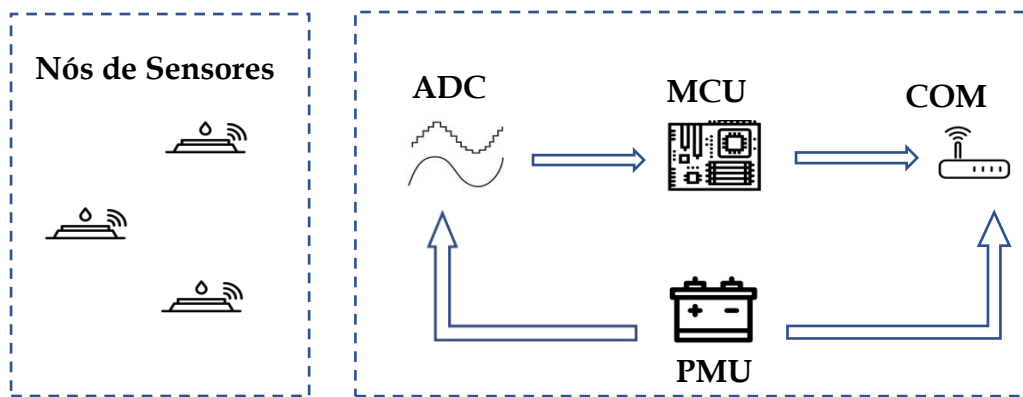


Figura 2.4-Arquitetura de Nó IoT

2.2.1. Microcontrolador (Microcontroller Unit -MCU)

O funcionamento base dos sistemas IoT requiere um dispositivo capaz de processar e armazenar (até certo ponto) os dados obtidos através da sua rede de sensores. Existe uma elevada variedade de microcontroladores disponíveis, sendo os dispositivos SoC (*System on a Chip*) a escolha mais seguida atualmente para este tipo de sistemas.

Neste capítulo serão analisadas as funcionalidades de algumas das placas disponíveis no mercado, nomeadamente o Arduino, o Raspberry Pi, o ESP8266 e o PSoC.

2.2.1.1. Arduino

O Arduino, projeto iniciado em 2005 em Itália, é uma plataforma *open-source* frequentemente utilizada para projetos de programação com base em circuitos eletrônicos. Desde o início da sua criação até ao momento da escrita foram já comercializados vários modelos da placa, tal como demostra a Figura 2.5, onde se encontram presentes o Arduino Uno, Mega, Due, Leonardo, Micro, Nano e Yun (esquerda para a direita, cima para baixo, respetivamente).

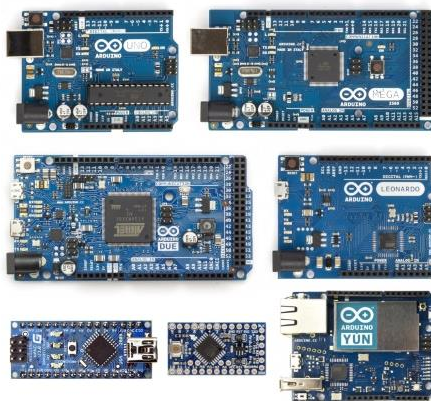


Figura 2.5- Placas Arduino. Adaptado de [6]

As principais vantagens das placas são a componente *open-source* da tecnologia, acessibilidade da linguagem de programação (linguagem mais comum é o C++), interoperabilidade entre várias plataformas bem como a existência do seu próprio ambiente de desenvolvimento “IDE” (*Integrated Development Environment*). As suas principais desvantagens são a sua reduzida memória e a incapacidade de correr um sistema operativo.

2.2.1.2. Raspberry Pi

O Raspberry Pi, lançado no Reino Unido em 2012, é um pequeno computador, munido de várias entradas periféricas e entradas I/O, capaz de suportar e correr diferentes sistemas operativos (sendo o mais usado o sistema Linux).



Figura 2.6- Raspberry Pi Modelo 3 [7]

Apesar das evidentes vantagens de utilização deste SoC como microcontrolador num sistema IoT existem ainda algumas desvantagens sobre o seu uso. As principais são a inexistência de um *clock* de tempo real (RTC) que pode provocar entraves em sistemas que necessitem de um controlo e/ou registo no momento exato e a ausência de qualquer conversor analógico-digital (ADC) embutido, tornando o sistema incapaz de receber diretamente sinais de sensores analógicos sem a existência de um módulo externo, ao qual se possa conectar.

2.2.1.3. PSoC

O PSoC, criado nos Estados Unidos da América em 2002 pela *Cypress Semiconductor*, foi dos primeiros exemplos de SOC's disponíveis no mercado.

A sua arquitetura consiste num *core* CPU, osciladores (interno e externo) e vários blocos, analógicos e digitais, configuráveis entre si. A quantidade de blocos analógicos e digitais disponíveis depende da versão de sistema utilizada.

Mais pormenorizadamente podemos distinguir, a nível analógico, conversores tanto analógico-digital como digital-analógico (ADC e DAC respetivamente), interruptores, filtros, comparadores e amplificadores operacionais[8]. Na gama digital encontramos vários temporizadores, contadores, moduladores (*PWM*) e interfaces de comunicação[9].

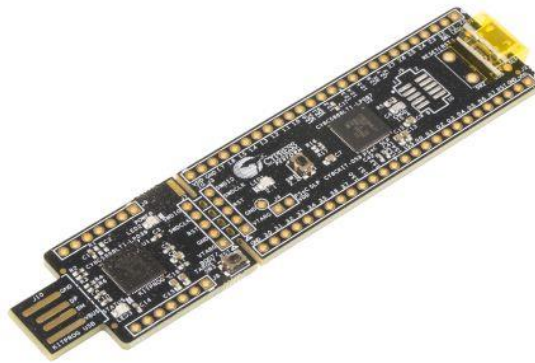


Figura 2.7- PSoC 5LP [10]

Para auxiliar o manuseamento dos vários blocos eletrônicos constituintes da placa, existe disponível um IDE próprio e intuitivo - *PSoC Creator* - que permite a utilização de *designs* gráficos para a programação dos circuitos existentes.

2.2.1.4. ESP8266

A família SoC ESP8266, produzida pela *Espressif Systems*, são módulos *low-cost* definidos, apesar de possuírem um microprocessador próprio, como uma solução de interface entre um microcontrolador pré-existente e uma rede *Wi-Fi*[11].



Figura 2.8- ESP8266 ESP-12E [12]

Desde o seu lançamento em 2014, encontram-se já disponíveis várias atualizações da placa. Todas elas, no entanto, possuem o mesmo processador (*32-bit Tensilica microcontroller*) sendo o fator diferenciador a quantidade e estilo dos pinos e a memória *flash* disponível.

Apesar das existentes vantagens de utilização em termos de conectividade é importante referir que, devido à necessidade de ser ‘alimentada’ a 3,3 V, a placa ESP8266 necessita de *hardware* adicional de maneira a poder ser utilizada em conjunto com outros módulos.

2.2.1.5. Análise comparativa

Como demonstrado pelos pontos anteriores, existem várias possibilidades de escolha de placas a utilizar para os nós que se pretendem implementar. Com vista à escolha dos equipamentos é importante ter em conta que o sistema tem a necessidade de ser *low-cost* e *low-power*.

De maneira a poder escolher os modelos mais viáveis a aplicar no projeto, foi elaborada a seguinte tabela de comparação:

Tabela 2.3- Comparação de MCUs

	Arduino Mega	Raspberry Pi 3	PSoC 5LP	ESP8266
Processador	ATmega2560	BCM2837 64bit CPU	32-bit ARM Cortex-M3 CPU	Tensilica L106 32-bit
Sistema Operativo	-	Linux	-	-
Linguagem Programação	C++ (IDE Arduino)	Qualquer	C (IDE PSoC Creator)	MicroPython Arduino IDE (C++)
Número de Pinos	100 pinos	40 pinos	52 pinos	28 pinos
Consumo	50mA	310 mA	15,4 mA	15 mA
Dimensões (mm)	53.3 x 101.52	56 x 85	11 x 2.4	24.5 x 14
Peso (g)	37	42	18	7
Preço (€)	≈ 32	≈ 44	≈ 8	≈ 8

2.2.2 Módulo de gestão de potência (*Power Management Unit* – PMU)

As PMUs são os elementos que controlam a distribuição de potência pelas diferentes plataformas digitais presentes nos sistemas eletrônicos. Estas são essencialmente constituídas por três componentes:

- LDO (*Low-Dropout Regulator*);
- Conversores DC/DC;
- Conversores de Nível Lógico.

2.2.2.1 LDO

LDO são circuitos que fornecem uma aplicação simples, rápida, de baixo custo e ruído que permitem regular a tensão de saída quando esta possui um valor muito próximo à tensão de entrada.

Utilizados para diminuir a tensão ($V_{in} > V_{out}$) é desejável, tendo em conta que a conversão é feita dissipando a diferença entre as duas tensões, que estas se encontrem próximas uma da outra. A utilização destes circuitos para conversões onde o rácio V_{in}/V_{out} é elevado leva a uma baixa eficiência energética bem como a limitações térmicas.

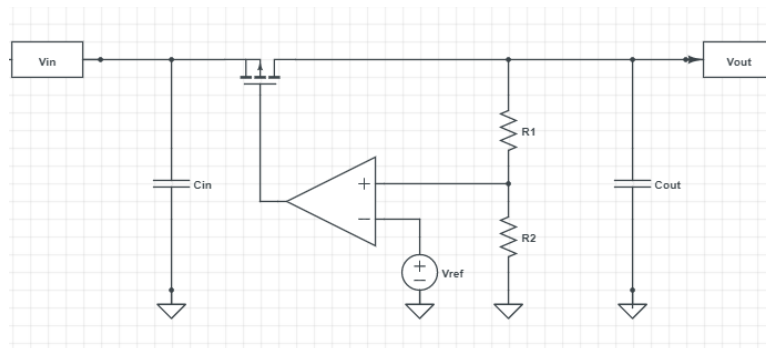


Figura 2.9- Arquitetura típica LDO

A arquitetura, apresentada na Figura 2.9, consiste num LDO ajustável (através do divisor resistivo) composto por um elemento resistivo (PMOS) controlado pela tensão de saída do comparador.

O seu funcionamento consiste na alteração da tensão da *gate* do transístor conforme a variação da tensão de saída (devido a um aumento ou diminuição da carga). Para assegurar uma tensão de *drive* da *gate*, LDOs têm normalmente uma tensão mínima de entrada $V_{in} >$

2,5 V. De maneira a permitir aplicações para tensões de entrada inferiores aos 2,5 V são utilizados LDOs com um MOSFET de canal N.

2.2.2.2. Conversores DC/DC

De forma a serem atingíveis conversões mais eficientes energeticamente, tanto para níveis de tensão superiores como para níveis de tensão inferiores, podem ser utilizados conversores DC/DC. Devido às características do sistema a projetar serão revistos, neste ponto, algumas das possibilidades de conversores *step-down* disponíveis no mercado para os níveis de tensão pretendidos.

Tendo em conta a necessidade de alimentação do módulo de comunicação (representado pela placa ESP8266) a 3,3V, será necessária a implementação de um conversor com esse valor de tensão de *output*.

2.2.2.3. Conversores de Nível Lógico

Dependendo da tensão utilizada na alimentação dos dispositivos eletrónicos assim variam também os limiares pelos quais são regulados os níveis lógicos utilizados na comunicação digital. Os níveis lógicos (tensão reconhecida pelo sistema como correspondente ao nível lógico binário 1 ou 0) são geralmente definidos em função das tensões máxima e mínima do dispositivo, sendo que ao existir comunicações entre dispositivos com diferentes alimentações é necessária uma conversão dos limiares existentes para que a informação seja transmitida corretamente.

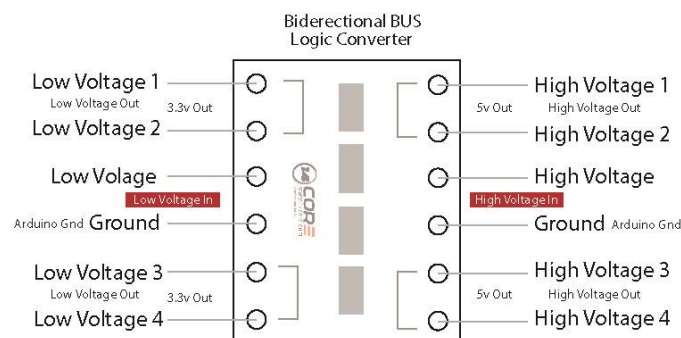


Figura 2.10- Esquemático Conversor de Nível Lógico [13]

A um conversor lógico, tal como apresentado na Figura 2.10, são conectadas as placas com as diferentes alimentações cujos valores se pretendem converter. A de maior tensão no lado direito (pino *High Voltage*) e a de menor no lado esquerdo (pino *Low Voltage*), sendo em ambas definido o valor do *ground* no seu *pin* específico. Após definidos os valores das tensões de alimentação são ligados aos restantes *pins* os valores de tensão a converter.

2.2.3. ADC (Analog-Digital Converter)

Os conversores analógico-digitais são considerados componentes críticos[14] não só em sistemas IoT mas em qualquer outro que necessite processamento de sinais. Sendo este o nó por onde o sinal proveniente dos sensores é primeiramente recebido, qualquer erro na conversão do sinal, que tenha como consequência perda e/ou transmissão de informação errada, resulta numa reduzida fiabilidade de medição e, por conseguinte, no sistema de monitorização.

Neste ponto far-se-á a avaliação dos módulos ADC mais comumente utilizados.

2.2.3.1. SAR (*Successive Approximation Register*) ADC

As arquiteturas SAR utilizam um algoritmo de comparação, apresentado na Figura 2.11 que consiste na comparação sucessiva entre o sinal de entrada (V_{in}) e a tensão de comparação do conversor (V_{DAC}). A tensão V_{DAC} é obtida à saída de um DAC (*Digital-Analog Converter*) de n -bits (mesmo número de bits do bloco ADC) alimentado à tensão V_{REF} . O início da conversão no bit mais significativo impõe à saída do DAC uma tensão inicial de $\frac{V_{REF}}{2}$.

Este algoritmo realiza a comparação bit a bit (do mais para o menos significativo), sendo que ADCs com maior resolução requerem automaticamente maior número de ciclos e, conseqüentemente, maior tempo de conversão[15].

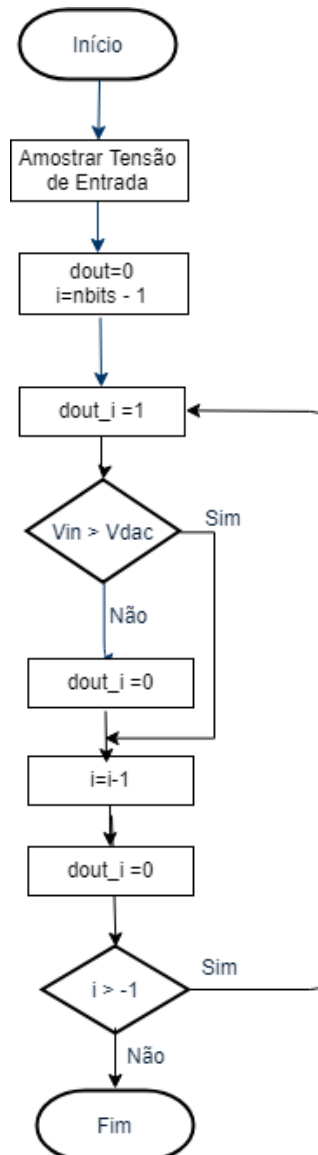


Figura 2.11- Algoritmo de Comparação SAR. Adaptado de apontamentos

2.2.3.2. ADC Sigma-Delta

A modulação sigma-delta é uma técnica que recorre a processamento analógico e digital do sinal, de maneira a melhorar a relação sinal ruído em relação a outros modelos de conversão.

Estes conversores utilizam o princípio de sobre-amostragem para repartir o ruído proveniente da quantização do sinal num mais alargado espetro de frequências.

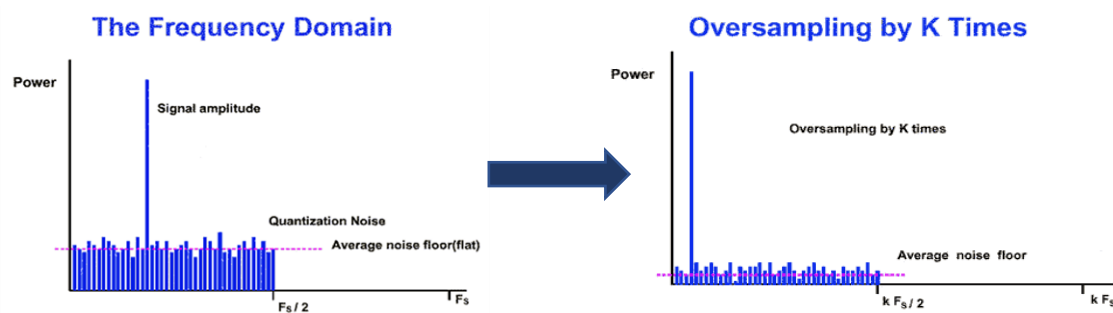


Figura 2.12- Efeitos da sobre amostragem no espectro da frequência. Adaptado de [16]

É ainda utilizado o mecanismo de *noise shaping* que, através do bloco integrador presente no conversor, redireciona o ruído existente para fora da largura de banda do sinal original (Figura 2.13).

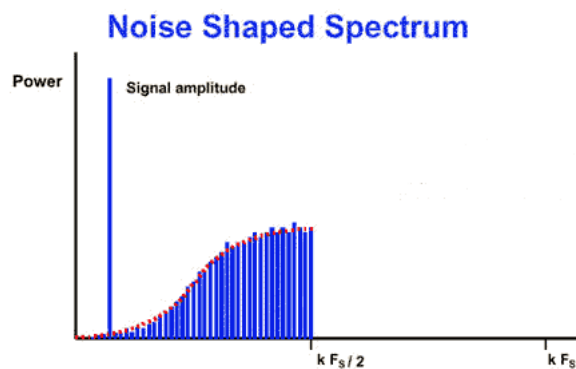


Figura 2.13- Noise Shaping [16]

2.2.4. Módulo de Comunicação

Sendo a principal função dos sistemas IoT a integração e conexão de diferentes sistemas, os protocolos de comunicação que possibilitam a troca de dados entre os diferentes componentes são uma parte fulcral do funcionamento destas tecnologias.

Neste capítulo serão analisados os protocolos mais comumente utilizados em sistemas IoT diferenciando entre protocolos locais (controladores-periféricos) e protocolos de rede.

2.2.4.1. Protocolos Locais

2.2.4.1.1. Universal Asynchronous Receiver and Transmitter (UART)

O UART é um circuito integrado capaz de lidar a conversão entre comunicação em série e em paralelo[17]. Apesar da comunicação em paralelo permitir uma maior velocidade de transmissão, também exibe, devido às ligações em comum, um maior risco de distorção e ruído. Ligações em série permitem, então, uma maior segurança e fiabilidade na comunicação.

O modo assíncrono no qual a transmissão é efetuada permite que o emissor não precise enviar o sinal de *clock*.

O protocolo usa um formato de *frame*, tal como se pode verificar na Figura 2.14, constituída por um *start bit*, *data bit*, *parity bit* e *stop bit*.

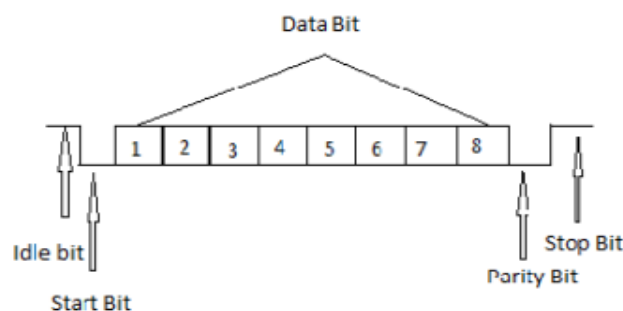


Figura 2.14- Formato de *Frame* UART [17]

2.2.4.1.2. Serial Peripheral Interface (SPI)

O SPI, introduzido em 1979 pela Motorola, foi primeiramente utilizado para conectar o microcontrolador aos seus periféricos. Existem, entre o *master* e o(s) *slave(s)*, quatro ligações:

- SLCK- Sinal de *clock* necessário para sincronização enviado pelo *master* a todos o(s) *slave(s)*;
- MOSI- *Master Out-Slave In*, Linha de transmissão de dados;
- MISO- *Master In-Slave Out*, Linha de transmissão de dados;
- SS_n - *Slave Select*, sinal utilizado pelo *master* para selecionar o *slave* específico.

De maneira a que a comunicação seja possível é necessário que esteja a ser utilizado o mesmo modo de comunicação, ou seja, que entre *master* e *slave* sejam aplicados o mesmo conjunto de parâmetros nomeadamente, frequência SCLK, polaridade do *clock* (CPOL) e fase do *clock* (CPHA).

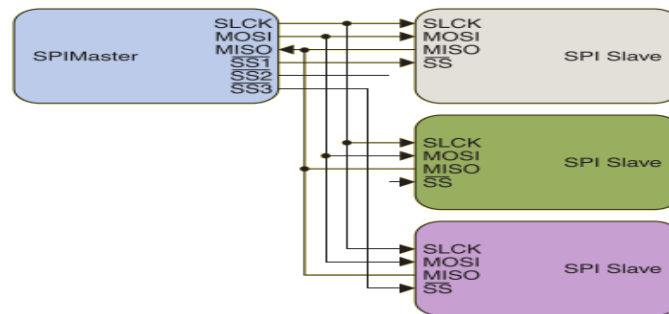


Figura 2.15- Topologia SPI[18]

Desenhado simplisticamente e para cumprir apenas as necessidades de comunicação, o SPI não define qualquer *data rate* máximo, qualquer reposta de confirmação de receção (ACK) ou controlo de fluxo.

2.2.4.1.3. Inter-Integrated Circuit (I^2C)

O I^2C , lançado em 1982 pela Philips, foi desenvolvido com a função de fornecer uma conexão simples entre o CPU e os seus periféricos numa televisão.

Este protocolo necessita apenas de dois cabos de ligação bidirecionais capazes de transmitir os sinais de *serial data* (SDA) e *serial clock* (SCL). Não é necessário a existência de sinal específico para a escolha do *slave*; o protocolo aceita a conexão entre vários *masters* e *slaves*. A comunicação é realizada com auxílio a um endereço de 7 bits atribuído a cada *slave*.

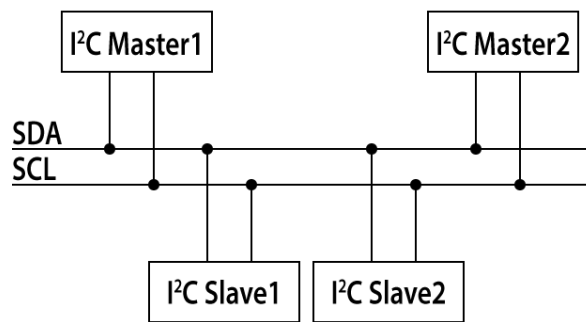


Figura 2.16- Topologia I2C [19]

O *master*, de maneira a poder efetuar a comunicação, emite a condição *START* de maneira a alertar os restantes dispositivos conectados com o bus. Posteriormente, é enviada a mensagem *ADDRESS* a definir qual o *slave* com o qual é desejado comunicar e, simultaneamente, qual é o modo de acesso aos dados (*Read* ou *Write*). O dispositivo cujo endereço coincide com o emitido pelo *master* emite uma mensagem de *ACKNOWLEDGE* que, ao ser recebida pelo *master*, o impele a iniciar a transmissão. Os dispositivos cujos endereços não coincidam aguardam o *terminus* da comunicação (indicado pela emissão da mensagem *STOP*).

2.2.4.2. Protocolos de Rede

2.2.4.2.1. IEEE 802.15.1 (Bluetooth)

O *Bluetooth* foi introduzido pela Ericsson. Esta, em 1998, criou, em consórcio com quatro outras empresas (IBM, Intel, Nokia e Toshiba), a SIG (*Special Interest Group*) com a finalidade de desenvolver uma tecnologia que permitisse interconectar dispositivos computacionais e de comunicação utilizando tecnologias sem fios[20].

Todos os dispositivos presentes numa rede *Bluetooth* possuem *hardware* idêntico. A definição entre *master* e *slave* ocorre no início de uma conexão onde, à unidade que a inicializa, é atribuída a designação de *master*. Os restantes dispositivos são definidos como *slaves* sendo-lhes atribuído um endereço temporário de 3 bits. Estas atribuições são válidas apenas enquanto a conexão estiver ativa.

Comunicação entre dispositivos *Bluetooth* pode ser dada por conexões *point-to-point* e *point-to-multipoint*. A rede formada pelo *master* e por um ou mais *slaves* chama-se *piconet*. Devido à reduzida dimensão dos endereços fornecidos aos *slaves* (3 bits) é apenas possível criar uma *piconet* no máximo com 8 dispositivos (1 *master* e 7 *slaves*).

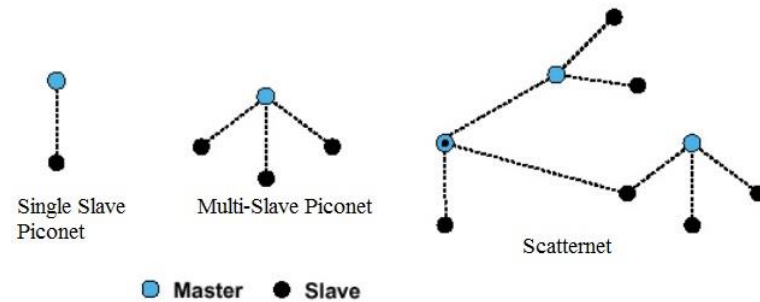


Figura 2.17-Topologias *Bluetooth* [21]

Existe a possibilidade de várias *piconets* se encontrarem conectadas através de nós de ligação (*master* ou *slave* que participa numa *piconet* diferente como *slave*) formando assim uma *scatternet*.

Tendo conhecido várias versões desde o seu lançamento foi lançado, em junho de 2016, o *Bluetooth 5.0*. Esta versão tem, como principal finalidade, o aumento da funcionalidade de sistemas IoT[22]. Esta versão tem, em relação ao seu precedente (*Bluetooth 4.2*), uma duplicação da largura de banda e uma quadruplicação do alcance (40m em ambientes interiores e 200m em ambientes exteriores).

2.2.4.2.2. IEEE 802.11 (*Wi-Fi*)

Wi-Fi é uma tecnologia baseada nos *standards* IEEE 802.11 que utiliza protocolos de comunicação bidirecionais para conectar dispositivos numa rede local sem fios (WLAN). As redes *Wi-Fi* possuem uma arquitetura por células chamadas de BSSs (*Basic Service Set*) que são constituídas por clientes e por APs (*Access Points*).

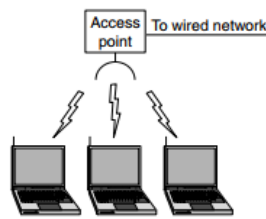


Figura 2.18- Rede Sem fios com *Access Point*[23]

O protocolo IEEE 802.11 suporta também ligações diretas entre os clientes, ou seja, sem a necessidade de comunicar com o *access point*, as chamadas ligações *ad hoc*.

De uma forma geral é definido o alcance das conexões *Wi-Fi* num raio de 30 metros em ambientes interiores e 150 metros para ambientes abertos. Estes valores são meramente indicativos pois qualquer obstáculo pode afetar os padrões estabelecidos.

2.2.4.2.3. IEEE 802.15.4 (ZigBee)

O *ZigBee*, criado em 2004, é um protocolo pertencente ao padrão IEEE 802.15.4 que define *standards* para uma arquitetura de baixo-custo e consumo[24]. A sua camada de rede suporta topologias em estrela, árvore e malha.

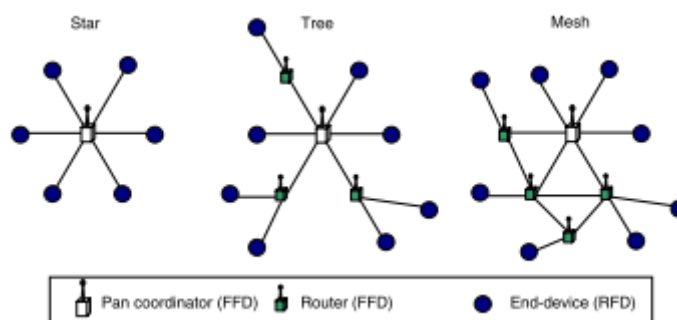


Figura 2.19- Topologias *ZigBee* [25]

Este protocolo identifica dois tipos de dispositivos. RFDs (*Reduce Function Device*) que correspondem a nós simples de comunicação capazes apenas de interagir com os nós coordenadores (FFD). Os nós FFD (*Full Function Device*) podem funcionar como os *routers* da rede sendo também os controladores da mesma.

O *ZigBee* fornece elementos de segurança tais como o estabelecimento e transporte de *security keys*, proteção de *frames* e gestão de dispositivos[26].

De maneira a evitar consumo desnecessário de energia são abordadas várias soluções capazes de prolongar o tempo de vida dos nós. Sendo o transceptor rádio o elemento que mais consome energia, é necessário encontrar alternativas ao modo “*always on*”. Coordenação entre a camada MAC e as camadas de mais alto nível (rede e aplicação) permitem adaptar a ativação do rádio eliminando o consumo desnecessário de energia durante os períodos de *idle listening*.

Os elevados elementos de segurança e reduzido consumo de energia conferem ao *ZigBee* elevado potencial no desenvolvimento de redes de sensores sem fios.



2.2.5. Módulo de Sensores

O elemento mais básico do sistema de monitorização é este que nos permite realizar a aquisição de dados do ambiente, o sensor. De acordo com as especificações do sistema e o objetivo para o qual está proposto, foram definidos como principais os dados relativos ao caudal e à pressão da água (tanto à saída como à entrada da bomba) bem como o valor da temperatura.

2.2.5.1. Sensor de Caudal

Tendo em conta que o sistema está dimensionado para um caudal de $16 \text{ m}^3/\text{h}$ ($\approx 4,44 \text{ l/s}$) serão apresentadas na Tabela 2.4 algumas das possibilidades encontradas:

Tabela 2.4- Sensores de Caudal

	 <p>Figura 2.20- Série FPR-200 [27]</p>	 <p>Figura 2.21- Série FTB8000B [28]</p>
Gama [m^3/h]	0,11 até 11,3	0,05 até 20
Output	4-20 mA, <i>Pulse</i> ou <i>Relay Output</i>	<ul style="list-style-type: none"> • Sem <i>Output</i> (FTB8015B); • <i>Reed Switch</i> (FTB8015B-PT); • Sensor Efeito <i>Hall</i> (FTB8015B-PR).
Preço	≈ 410€	≈ 465€

Uma alternativa aos sensores comercializados é a utilização do próprio *hardware* do sistema para obter os valores necessários.

A ação mecânica da bomba em modo turbina do próprio sistema que, ao se encontrar afetada pelo fluxo de água, atinge uma velocidade proporcional à velocidade do fluido. O número de rotações da turbina coloca o rotor em movimento produzindo um sinal do qual é possível extrapolar o valor do caudal.

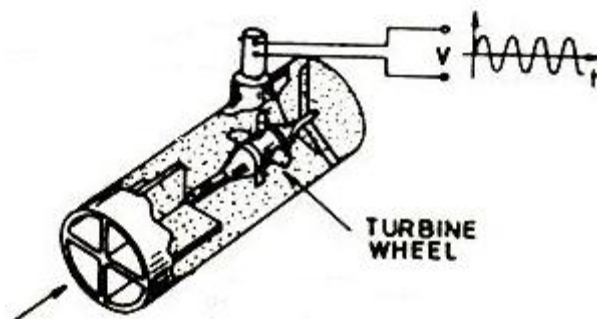




Figura 2.22- Funcionamento básico do medidor. Adaptado de [29]

2.2.5.2. Sensor de Pressão

Apesar do sistema estar preparado para funcionar à pressão de 10 bar (1Mpa) de acordo com os parâmetros atuais é expectável que o valor atingido seja de apenas um décimo do indicado. Tendo em conta os sensores encontrados foi elaborada a tabela de comparação apresentada seguidamente.

Tabela 2.5- Sensores de Pressão



	 Figura 2.23- Série PXM209 [30]	 Figura 2.24- SEN0257 [31]
Gama [Mpa]	até 1	até 1,6
Output	2 fios, 4 a 20mA	3 pin interface, $\leq 2,8mA$
Preço	265€	≈ 10€

2.2.5.3. Sensor de Temperatura

A indicação da temperatura, apesar de não ser fulcral para o funcionamento do sistema, constitui uma informação útil sobre o seu bom ou mau funcionamento. Devido às características do sistema serão monitorizados os valores de temperatura tanto da água como do ambiente envolvente.

Tendo em conta a localização do sistema, bem como as variações sazonais da temperatura será considerado que os valores obtidos estarão compreendidos entre os 0 e os 40 °C.

Tabela 2.6- Sensores de Temperatura

	 <p>Figura 2.25- ds18b20 [32]</p>	 <p>Figura 2.26- MNS-9-W1-TS-WT-L03 [33]</p>
Gama [°C]	-55°C a 125°C	-40°C a 100°C
Output	1 wire interface	Wireless Frequência = 900MHz
Preço	≈ 10€	≈ 56€

Arquitetura e Implementação do Sistema

Partindo das especificações para os sistemas IoT descritas no capítulo anterior e, mais concretamente, na arquitetura base presente na Figura 2.4, foi dimensionada uma arquitetura própria capaz de se adequar às características do sistema PaT ao qual o nó de monitorização irá ser aplicado. Neste capítulo procederemos à descrição da conceção e implementação da arquitetura dimensionada.

A implementação do projeto pode ser dividida em quatro grandes fases:

- Obtenção dos dados dos sensores (PSoC);
- Transmissão dos dados para o servidor (ESP);
- Criação da base de dados;
- Montagem do nó de monitorização no sistema PaT.

3.1. Nó de agregação

De entre os microcontroladores estudados foi escolhida a placa PSoC CY8CKIL-059 5LP Prototyping Kit.

Este elemento da arquitetura, sendo a primeira interface com os nós de sensores, necessita possuir um número suficiente de pins (tanto digitais como analógicos) para receber as informações do ambiente circundante bem como para comunicar com os restantes nós da arquitetura.

A placa escolhida destaca-se das restantes por cumprir todas as necessidades tanto ao nível da receção de dados como ao nível do seu tratamento pelos blocos analógicos e digitais, pela área reduzida de ocupação e, mais importante, pelo menor consumo energético de todas as placas estudadas.

3.1.1. Medição de temperatura

Foi escolhido, com vista a recolher os valores de temperatura ambiental e da água, o sensor DS18B20.

Sensor digital, capaz de enviar e receber informação proveniente do controlador através de uma interface *one-wire*. Apesar de possuir três *pins* de ligação (V_{DD} , GND e DQ) o sensor pode ser alimentado através da energia que deriva do próprio *pin* de comunicação (DQ) sendo, portanto, opcional a existência de uma fonte de alimentação externa.

Devido à existência de números de série únicos, o DS18B20 permite a existência de vários sensores no mesmo bus de dados (Figura 3.1). Considerando a possibilidade de existir diferentes tipos de medições de temperatura (ambiental e da água) em diferentes zonas do sistema, a utilização de um único *bus* poderia ser vantajosa para diminuir a complexidade da montagem do protótipo.

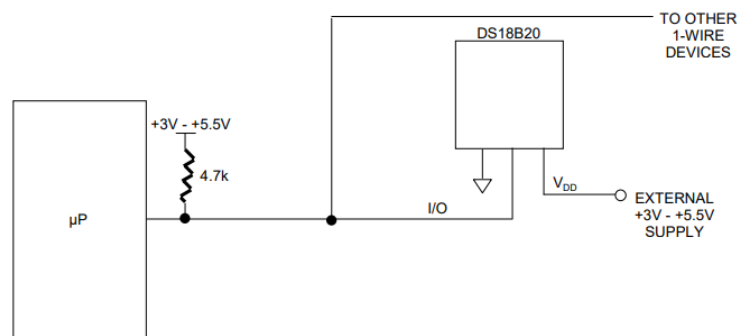


Figura 3.1- Ligação entre DS18B20 e microcontrolador [37]

Contudo, apesar da possibilidade de os sensores utilizados permitirem a utilização do protocolo I2C (capítulo 2.4.1.3.), a disponibilização de um bloco pré-existente no IDE do *software* (DS18B20_reader na Figura 3.2), impossibilita a utilização desse mesmo protocolo. O bloco em causa força que o seu número de entradas (N) seja igual ao número de pins de entrada utilizados no PSoC o que, obrigatoriamente, impede a existência de apenas um fio para comunicação.

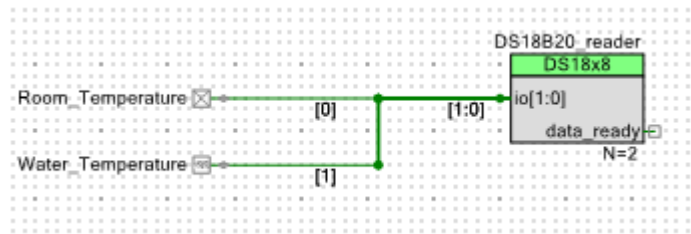


Figura 3.2- Circuito de obtenção de valores de temperatura

O bloco (DS18x8 Digital Temperature Sensor no *datasheet* da Cypress) está implementado como uma máquina de estados[34] maioritariamente controlada por *software*.

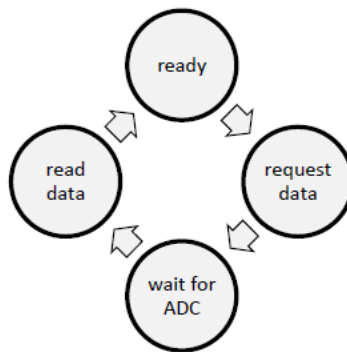


Figura 3.3- Estados do bloco DS18x8 [34]

O envio e receção dos *interrupts* necessários para a mudança entre os estados da Figura 3.3 é assegurado pelo bloco DS18x8, de acordo com a chamada das funções implementadas no código abaixo exposto; “*SendTemperatureRequest()*” limpa os *buses* de informação e desencadeia o início da conversão por parte dos sensores. Estando a conversão completa, os sensores emitem a *flag* “*DataReady*” permitindo ao bloco ler as temperaturas (ambiente e da água) guardando-as respetivamente nas variáveis *room_temperature* e *water_temperature*.

3.1.2. Medição de pressão

Foi escolhido, com vista a recolher os valores de pressão nos diversos pontos do sistema, o sensor SEN0257.

O dispositivo, alimentado a 5V, é capaz de suportar pressões até os 2 MPa sendo que a gama de medições varia entre os 0 e os 1,6 MPa. A sua ligação é feita com uma interface de três pins V_{DD} , GND e sinal.

O processamento do sensor emite, à saída, um sinal analógico que cresce linearmente com a pressão aplicada no sensor (Figura 3.4).

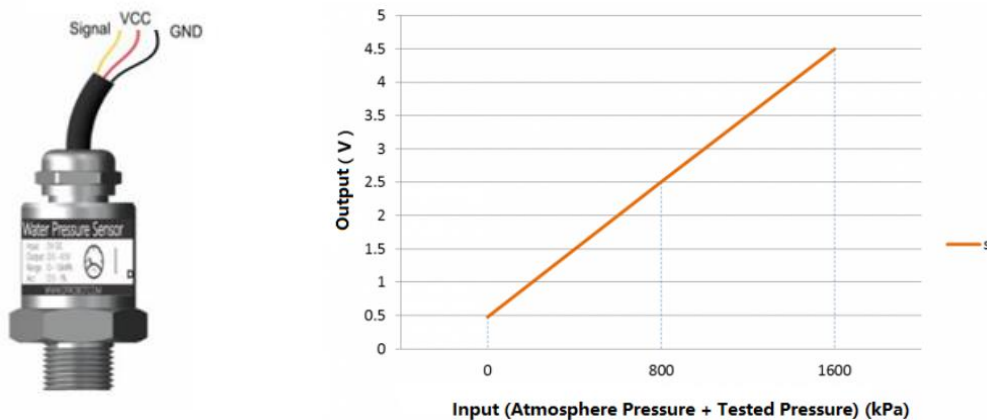


Figura 3.4- Característica SEN0257. Adaptado de [35]

Sendo os sensores de pressão escolhidos analógicos o principal bloco ser utilizado é o conversor analógico digital. Na montagem projetada (Figura 3.5), é utilizado um amplificador operacional em montagem seguidora como *buffer* de entrada do ADC de maneira a assegurar uma maior impedância de entrada.

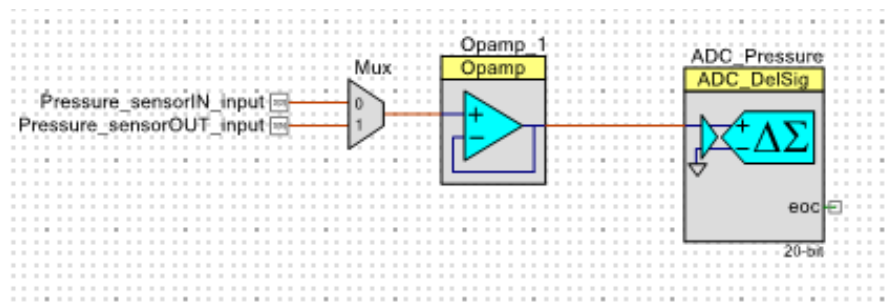


Figura 3.5- Circuito de obtenção dos valores de pressão

Devido à existência de apenas um conversor Sigma-Delta e, de maneira a uniformizar as conversões para ambos os sensores de pressão, é partilhada a entrada do ADC por ambas as entradas provenientes dos sensores, sendo estas comutadas pelo *multiplexer*.

O multiplexer utilizado (bloco 'AMux' no *datasheet* da Cypress) é controlado por *software*, permitindo uma gestão mais eficaz da comutação de entradas, que ocorre apenas quando termina a obtenção e conversão do valor de cada um dos sensores.

Os valores de pressão de entrada e saída ficam armazenados nas variáveis *pressureIN* e *pressureOUT* respetivamente. O algoritmo implementado encontra-se exposto na Figura 3.6.

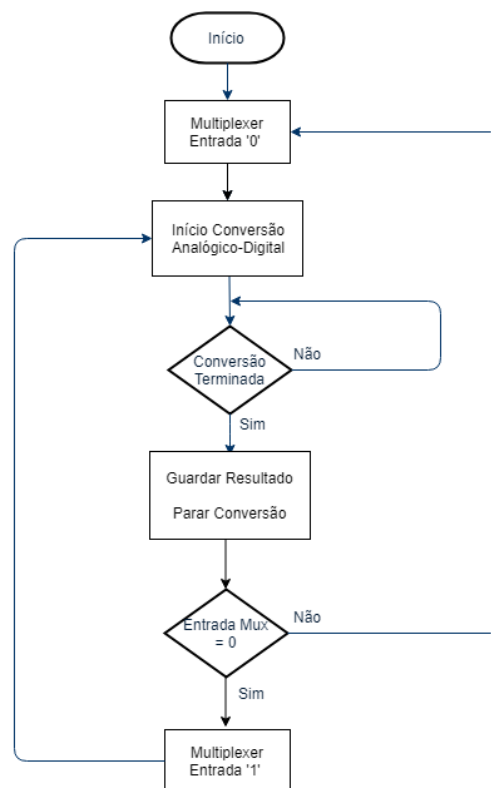


Figura 3.6- Fluxograma do algoritmo de medição implementado

3.1.3. Medição do caudal

Foi escolhido, com vista a recolher o caudal do sistema, o sensor YF-DN50. Possui uma interface de três *pins* V_{DD} , GND e sinal, sendo a sua alimentação realizada a 5V.

O dispositivo suporta pressões até os 1,75 MPa e sustém medições válidas para caudais até os 350 L/min.

O sensor produz, à saída, uma onda quadrada, de frequência variável mediante o caudal observado. A característica, observada na Figura 3.7, pode ser obtida pela equação 3.1.

$$f = 0,2Q \quad (3.1)$$

Onde f equivale ao valor da frequência (Hz) e Q ao caudal (L/min).

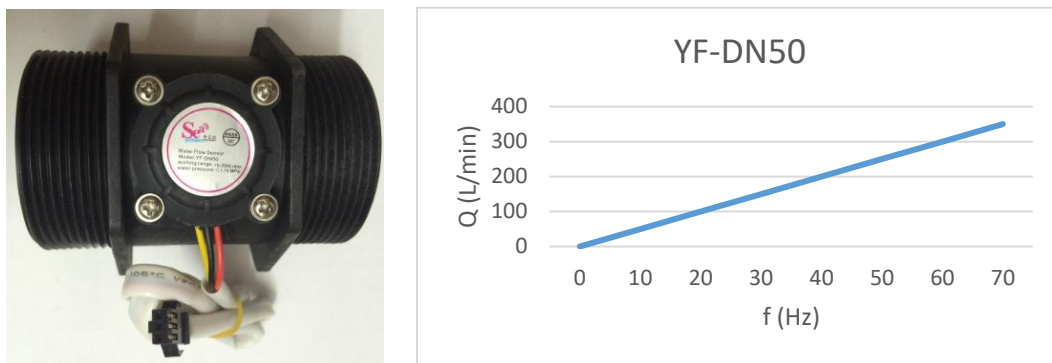


Figura 3.7- Característica YF-DN50. Adaptado de [36]

O sensor de caudal utiliza um transdutor que varia, à saída, a frequência do seu sinal. Como tal, no nó agregador, foi necessário utilizar uma montagem capaz de contar as oscilações do sinal proveniente do sensor e, posteriormente, com base no valor obtido, efetuar o cálculo final do fluxo de água.

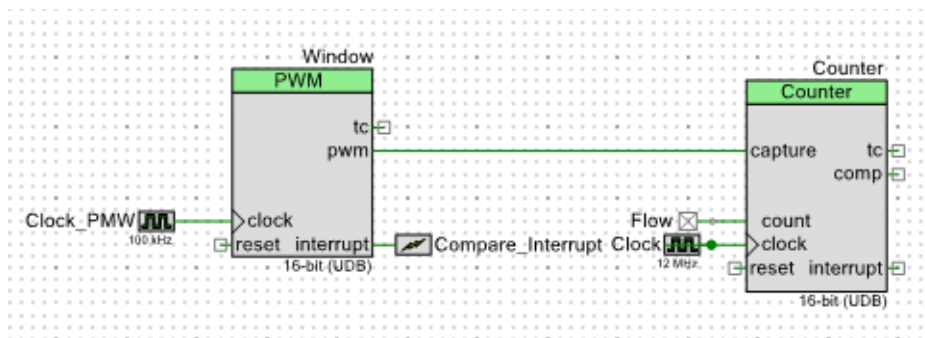


Figura 3.8-Circuito de obtenção do valor do caudal

De maneira a realizar o pretendido foi então elaborado o esquemático exposto na Figura 3.8. Este é constituído pelos blocos *PWM* (*Pulse Width Modulator*) e *Counter*, sendo que o sinal a ser analisado dá entrada através no pin “Flow”.

O bloco *PWM* define a janela de medição estabelecendo uma onda quadrada de *duty-cycle* 50% com um período de 500ms. Este sinal dá entrada no pin “Capture” do bloco *Counter* cujo modo de captura se encontra configurado como “*Falling Edge*”, reiniciando o contador no final de cada período da onda quadrada.

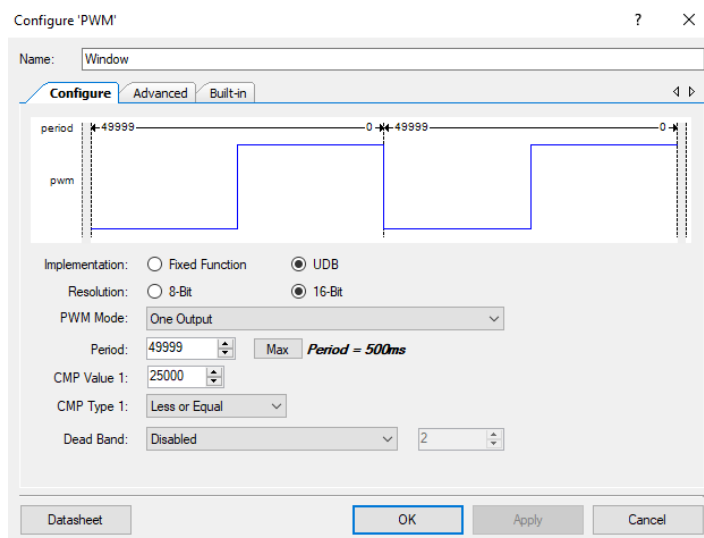


Figura 3.9- Janela gerada pelo bloco *PWM*

O número de oscilações obtido durante o período do sinal é multiplicado por 2 (de maneira a perfazer o valor da frequência) e guardado na variável *freq*. Dispondo da frequência do sinal é então aplicada a equação 3.1 obtendo-se finalmente o valor do caudal.

3.1.5. Envio da informação

O envio da informação recolhida para o nó de comunicação é realizado através do módulo UART (Figura 3.10) que assegura a comunicação série entre as duas placas. O módulo foi configurado para o *baud rate* de 115200 bps.

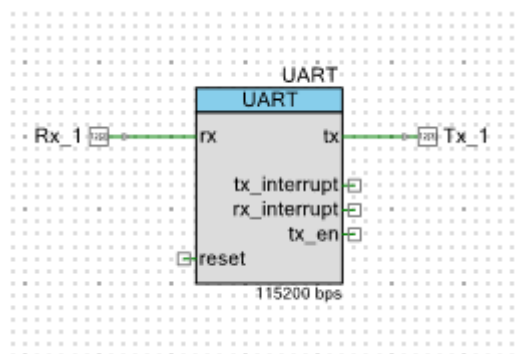


Figura 3.10- Módulo UART

Para se proceder ao envio é produzida uma *string* em notação *JSON* (de maneira a facilitar a leitura e armazenamento nos pontos seguintes do sistema) onde são incluídas as variáveis previamente preenchidas.

O código detalhado, elaborado para o nó de agregação, encontra-se incluído em anexo.

3.1.5. Disposição dos pinos

O planeamento tanto dos blocos a utilizar como dos *pins* I/O a atribuir, estão diretamente correlacionados com os sensores escolhidos para efetuar as medições. Como demonstram os pontos anteriores, o sistema possui sensores tanto analógicos como digitais. Na Figura 3.11 encontra-se representada a distribuição dos pinos utilizada no âmbito deste projeto.

A disposição implementada teve em conta não só as necessidades relativamente ao tipo de sensor, mas também à implementação do protótipo distribuindo os pinos relativamente aos sensores de um lado e, do outro, os pinos de ligação às camadas superiores do sistema

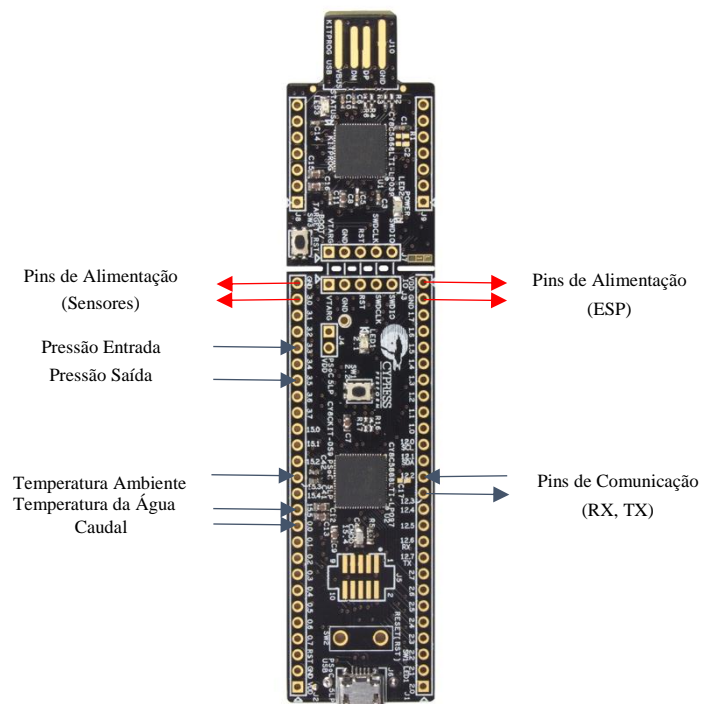


Figura 3.11- Distribuição dos pinos na placa PSoC. Adaptado de [38]

3.2. Nó de Comunicação

A ligação entre o nó de monitorização e o servidor será realizada pela placa Feather HUZZAH ESP8266. A integração do protocolo TCP/IP presente nos módulos ESP permite, melhor que qualquer outra placa, acesso a uma rede *Wi-Fi*.

As características desta família de microcontroladores adequam-se ao sistema a ser implementado devido à reduzida área ocupada e ao seu reduzido consumo energético (cumprindo os requisitos de *low-power*).

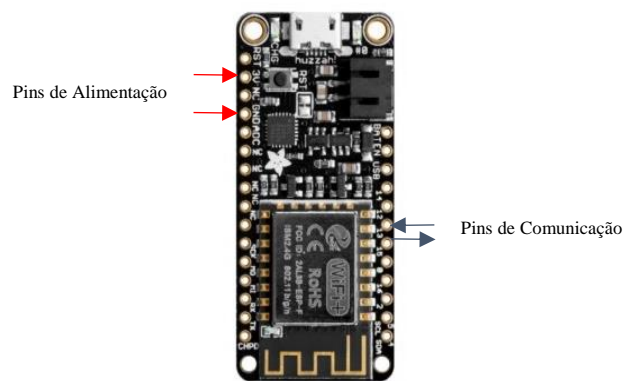


Figura 3.12- Distribuição de pins na placa ESP8266. Adaptado de [39]

O modelo em específico foi escolhido devido a uma maior facilidade de integração (disponibilidade de *pins*) e de programação (utiliza como ambiente o IDE *Arduino* sendo possível a compilação diretamente pela porta micro USB presente na placa).

Os *pins* específicos dedicados à comunicação série (RX e TX) presentes no módulo apresentam complicações quando são utilizados para efetuar a comunicação pretendida. Estes *pins* estão conectados ao conversor USB-Série usado para programar o dispositivo; dessa maneira, ao serem utilizados estes *pins* para comunicar com o PSoC, o código previamente carregado no dispositivo era substituído pelo *stream* de dados proveniente do PSoC.

Assim, é necessário a definição, por *software*, de novos *pins* dedicados à comunicação. A biblioteca "*SoftwareSerial*" permite a declaração necessária utilizando quaisquer pinos I/O existentes na placa. Para os objetivos necessários de transmissão e recepção foram então escolhidos os *pins* 12 e 13 correspondentes respetivamente a RX e TX.

O módulo ESP recebe, caracter a caracter, a informação enviada do nó agregador, concatenando-a na *string data* enquanto não for recebido o caracter de término da *string* enviada (designado como sendo '?'). Após a *string* se encontrar completa esta é guardada em *data-ToSend*, pronta a enviar quando a ligação *Wi-Fi* estiver ativa. A redundância utilizada (duas variáveis ao invés de enviar *data* diretamente) permite obter maior segurança especialmente em alturas de maior flutuação da rede *wireless*.

São utilizadas, de maneira a comunicar com o servidor, duas outras bibliotecas.

- ESP8266WiFi.h;
- ESP8266HTTPClient.h.

A primeira permite, através dos valores de *ssid* (*service set identifier*, o nome da rede) e da sua *password*, efetuar uma conexão *Wi-Fi* com a rede cujos parâmetros foram definidos.

A segunda biblioteca disponibiliza a utilização dos métodos existentes no protocolo HTTP (*Hypertext Transfer Protocol*). Sabendo que o ESP irá ser utilizado como um cliente que envia informações para um servidor, o único método que é utilizado é o *POST*.

Após criar uma variável da biblioteca enunciada, é utilizada a função *begin* de maneira a iniciar a comunicação. Esta função tem como parâmetros de entrada o URL do servidor composto pelo seu endereço IP na rede, pelo porto de entrada e pelo método a chamar aquando a entrada de informação.

De seguida, imediatamente antes do envio, é definido no cabeçalho da mensagem que os dados a enviar se encontram no formato JSON ("Content-Type","application/json").

O código utilizado na programação do dispositivo encontra-se exposto em anexo.

3.3. Conversor DC-DC

Como foi dito anteriormente, o módulo ESP é apenas capaz de sustentar, sem dano para os pinos, uma alimentação de 3.3V. De maneira a permitir a sua alimentação por parte do PSoC e assim evitar uma segunda fonte de alimentação, é necessário realizar a conversão 5V-3.3V entre os *pins* de alimentação de ambos os módulos.

Para esse efeito foi escolhido o regulador de tensão da família LD1117.

3.4. Conversor de Nível Lógico

A diferença entre os valores de alimentação das duas placas reflete também problemas no momento da comunicação; a informação na sua origem (PSoC) vem codificada com o valor de 5V para o valor binário '1' (*high*).

O conversor (Figura 3.13), recorrendo aos níveis lógicos LV/HV (correspondentes respetivamente a *low* e *high voltage*), varia o valor da saída em conformidade com as tensões e níveis lógicos apresentados.

Apresentando-se o nível lógico '1' na entrada HV1 existirá então uma tensão de 5V no dreno do transístor. A tensão na *source* começa a subir até atingir os 3.3V da *gate* altura em que o MOSFET deixa de conduzir ($V_{GS} = 0$).

Apresentando-se o nível lógico '0' na entrada HV1 o dreno do transístor ficará a 0V. Assim o diodo de substrato, forçando uma diferença de potencial de aproximadamente 0.7V, baixa a tensão da *source* até 0.7V causando um V_{GS} suficientemente elevado para o transístor voltar a conduzir levando a tensão de saída a tender para 0V.

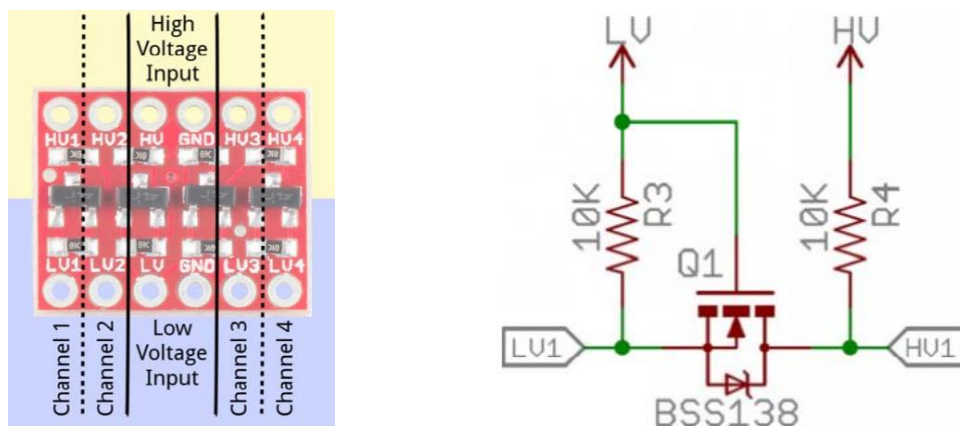


Figura 3.13- Conversor de nível lógico bidirecional. Adaptado de [37]

3.5. Elaboração e implementação do protótipo

Para a implementação final do nó de monitorização é utilizada uma placa furada.

Como foi referido no ponto 3.1.5., os *pins* do PSoC a utilizar foram distribuídos de maneira a considerar esta placa como o componente central da implementação final, dividindo as ligações aos sensores de um lado e as ligações ao ESP do outro.

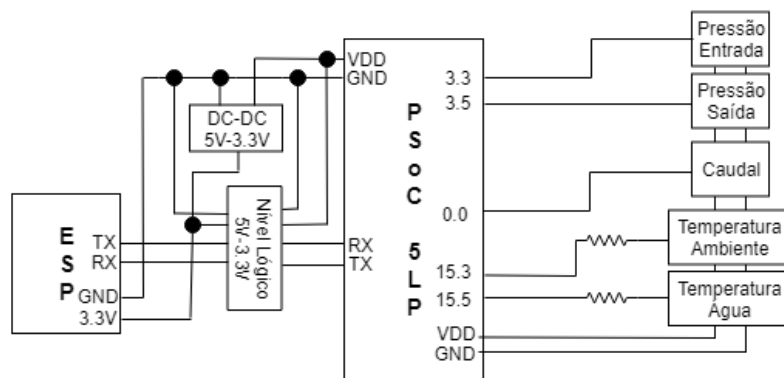


Figura 3.14- Esquemático da implementação

O ESP é alimentado a 3.3V através da conversão assegurada pelo módulo DC-DC (5V-3.3V). Os pinos definidos para comunicação requerem uma conexão cruzada (RX-TX/TX-RX), que se dá através do conversor de nível lógico utilizado.

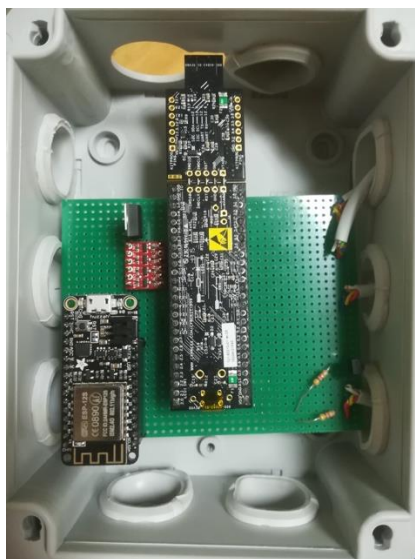


Figura 3.15- Protótipo utilizado no sistema

Após a elaboração do protótipo foi efetuada a montagem dos sensores no sistema físico e feitas as respectivas ligações ao nó de monitorização.

Os sensores de temperatura (ambiente e da água) não necessitaram, devido à sua disposição, de *hardware* adicional. O sensor de temperatura ambiente foi implementado, como

se verifica na figura anterior, diretamente na placa furada (canto inferior direito) enquanto o sensor de temperatura da água foi introduzido, de maneira a evitar alterações adicionais no sistema hídrico, no depósito de água.

As tubagens do sistema hídrico no qual os sensores ficarão montados são de dimensão DN50 existindo, portanto, em alguns casos, a necessidade de adaptar as secções das tubagens às dos sensores. Nas Figura 3.16 e Figura 3.17 são apresentadas as soluções implementadas para a montagem dos sensores de pressão e caudal, respetivamente.

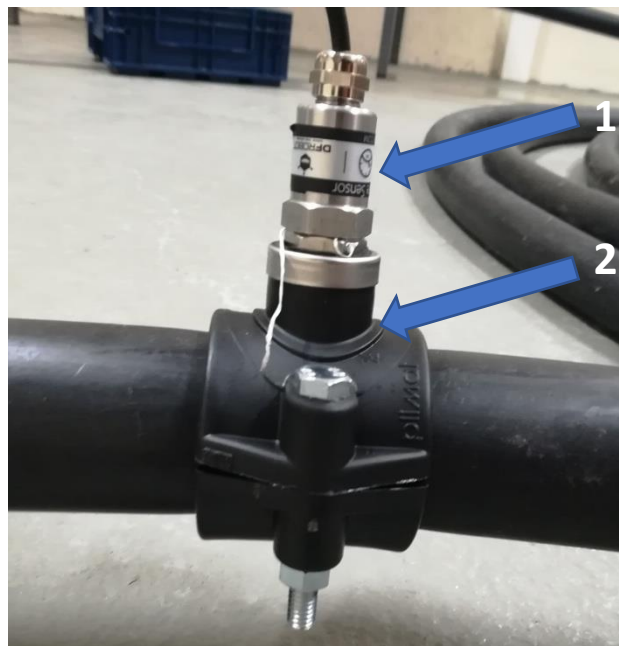
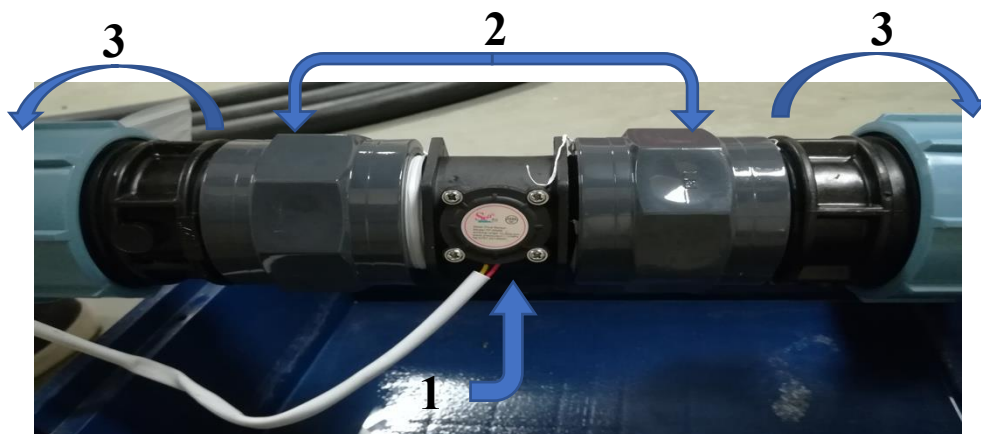


Figura 3.16-Montagem sensor de pressão: 1. Sensor, 2. Anel Adaptador DN50



**Figura 3.17- Montagem do sensor de caudal: 1. Sensor, 2. Adaptador M50 2'',
3. Adaptador Macho**

3.6. Base de Dados e Servidor

Para criar a base de dados necessária para suportar os dados recolhidos do sistema será utilizado o *software open-source* MongoDB (versão 3.6.2).

As bases de dados NoSQL no qual se integra o MongoDB permitem uma maior liberdade e flexibilidade na organização da informação em relação aos sistemas relacionais *MySQL*. Estas características poderão, no futuro, revelar-se importantes caso venha a surgir a necessidade de extensão da base de dados.

O *software* utilizado comporta também a utilização de linguagens de programação orientadas a objetos; este ponto permite que os dados recebidos no servidor em formato JSON (*JavaScript Object Notation*) sejam mais facilmente distribuídos pelos modelos criados.

Sendo a informação ser recebida em formato JSON, foi escolhido o *JavaScript* como linguagem de programação para a *back-end*; desta maneira, para definir o servidor, foi escolhido o ambiente Node.js. O uso desta plataforma é transversal aos sistemas operativos mais utilizados (*Windows, Linux, Unix, Mac OS*) possibilitando a operação, à semelhança do *software* utilizado para a base de dados, com a linguagem de implementação *javascript*.

O Node.js, baseado no Google V8, é um ambiente capaz de executar processos acionados por eventos I/O. Sendo uma tecnologia *open-source* existe um repositório de módulos, NPM (*Node Package Manager*), disponível para utilizar. Neste projeto foi utilizada a versão mais atual existente de momento, versão 9.4.0.

Os ficheiros *Node.js* criados contêm, então, os métodos a serem executados quando são acionados os eventos de receção de informação. Como se pode verificar no código utilizado na plataforma ESP, o envio da informação recolhida é feito para o porto 10010 (onde se encontra a correr o servidor) ativando o método *post*. É então, neste método, que é efetuado o armazenamento dos dados recebidos.

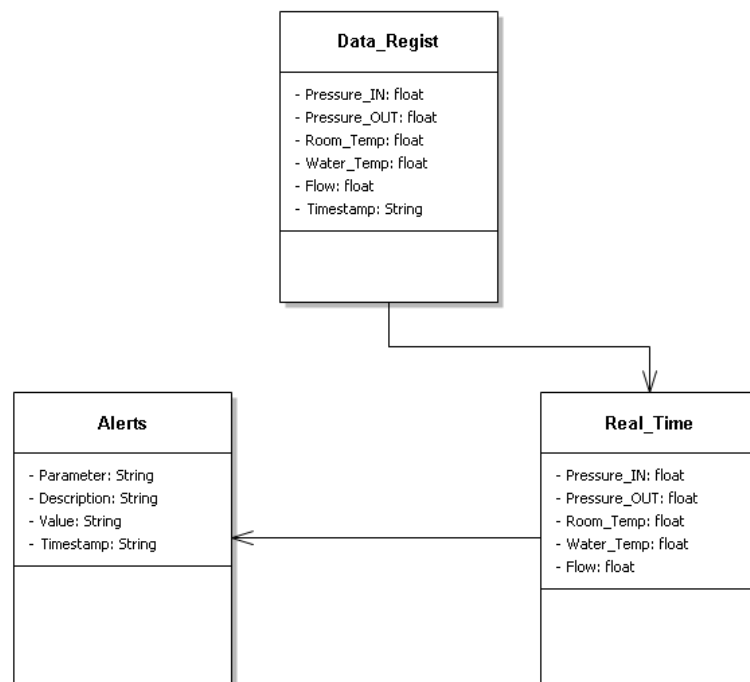


Figura 3.18- Esquema relacional da base de dados

O esquema relacional projetado para o sistema encontra-se exposto na Figura 3.18. O modelo organizacional base é representado pela tabela *Data_Regist* onde são guardados diretamente os objetos enviados pelo ESP. Este modelo possui, portanto, todos os parâmetros recolhidos do sistema (com a adição de um *timestamp*) guardando-os de trinta em trinta segundos (período de envio da informação por parte do nó de monitorização). O *Data_Regist* tem então como objetivo o contínuo armazenamento de maneira a permitir realizar um estudo do sistema temporalmente mais abrangente.

De maneira a separar o estudo temporal da visão momentânea dos parâmetros do sistema, foi criada um modelo ‘alimentado’ pela tabela anterior, de apenas uma entrada, onde estarão disponíveis os valores em tempo real (*Real_Time*).

A tabela *Alerts* ficaram registados, como o seu nome indica, os momentos em que os valores obtidos se encontrem fora dos padrões delimitados como normais para o sistema. Será criada uma entrada do tipo *Alerts* quando os valores ultrapassam os respetivos máximos

permitidos pelos sensores ou quando, havendo uma eventual falha na obtenção e/ou transmissão dos dados por parte do nó de monitorização, dê entrada no servidor uma mensagem com parâmetros nulos.

Esta tabela permite alguma agilidade na percepção de problemas no sistema, evitando a necessidade de analisar todos os dados recebidos do tipo *Data_Regist*.

A distribuição e armazenamento da informação na base de dados segue o fluxograma representado na Figura 3.19.

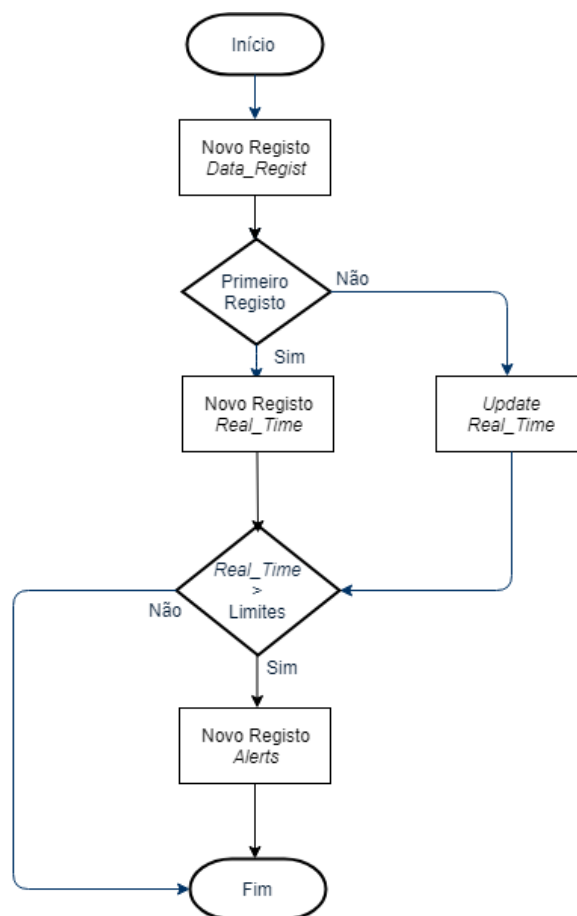


Figura 3.19- Fluxograma do algoritmo de armazenamento implementado

3.7. Interface gráfica

De maneira a facilitar a visualização dos dados obtidos em tempo real foi criada, com recurso ao *software Matlab*, a interface exposta na Figura 3.20.

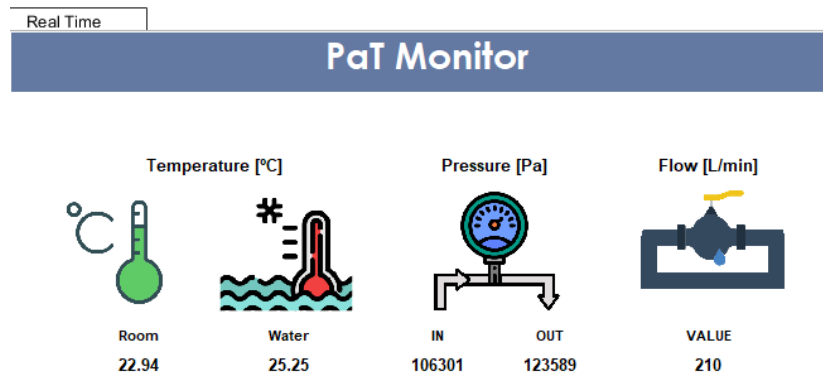


Figura 3.20- Interface Matlab

A informação do sistema exposta na interface é obtida a partir de um ficheiro *csv* escrito a partir do código do servidor. O código redigido permitirá a adição posterior de separadores adicionais ao existente (*Real Time*) simplesmente adicionando o nome desejado para o novo separador na variável '*TabNames*' e adicionando um novo painel na *guide*.

Resultados Experimentais

Neste capítulo serão apresentados os resultados das medições efetuadas no sistema (Figura 4.1) para dois modos distintos de funcionamento:

- *Bypass*- Obtido fechando as válvulas à entrada e saída da PaT (3 e 4 respetivamente), mantendo as restantes abertas. Neste modo apenas entram em consideração os valores da temperatura, caudal e pressão de entrada;
- PaT- Sistema completo (válvula 2 fechada e restantes abertas).

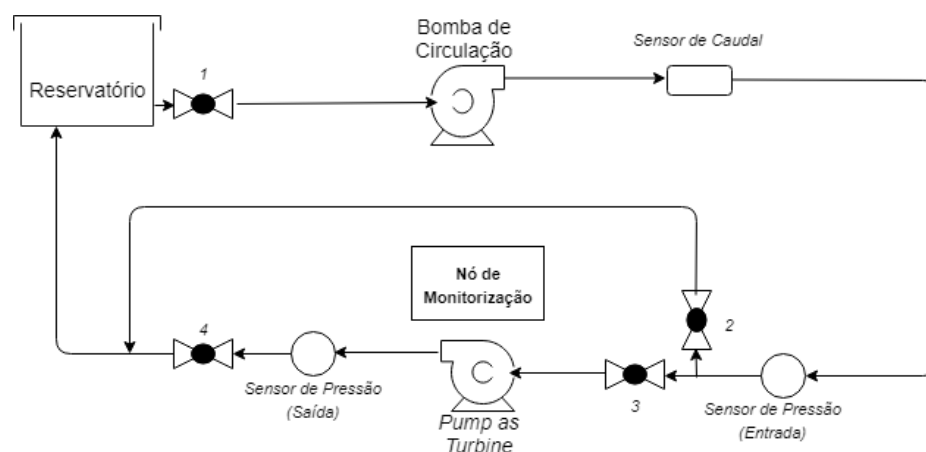


Figura 4.1- Esquema detalhado da montagem

Em primeiro lugar, antes de efetuar qualquer medição, é necessário atestar a segurança do sistema. Abrindo a válvula do depósito (válvula 1), enchemos os tubos com água confirmando que não existe qualquer fuga nos locais em que os sensores foram introduzidos no sistema. Esta primeira verificação é realizada sem qualquer bombeamento sendo que a pressão é reduzida.

Em cada modo de funcionamento foi então realizado o teste em pressão comprovando a solidez da montagem.

4.1. *Bypass*

No modo *bypass* é apenas variada a alimentação da máquina que bombeia a água pelo sistema; para tal recorremos a uma central auxiliar de proteção e ao autotransformador trifásico da Figura 4.2.



Figura 4.2- Autotransformador trifásico

Como é possível verificar na imagem, o autotransformador é regulado em percentagem da sua tensão máxima. De maneira a reduzir a corrente de arranque da bomba de água, são aplicadas pequenas variações de tensão no autotransformador sendo que, após a bomba de água ter desenvolvido binário suficiente, se pode variar mais rapidamente a tensão aplicada aos seus terminais.

Foi escolhido, como ponto central para as medições, o valor de 80% da tensão máxima do autotransformador. Este será a base a partir do qual serão realizadas as variações no sistema.

A medição do funcionamento em *bypass* apresentada nas Figura 4.3, Figura 4.4 e Figura 4.5 seguiu os seguintes passos:

1. Máquina desligada;
2. Tensão aplicada (80% no autotransformador) - Duração, 10 minutos;
3. Redução da tensão (60% no autotransformador) – Duração, 2,5 minutos;
4. Redução da tensão (50% no autotransformador) – Duração, 2,5 minutos;
5. Aumento da tensão (80% no autotransformador) – Duração, 4 minutos;
6. Redução da válvula de *bypass* (válvula 2) mantendo tensão – Duração, 3 minutos;
7. Normalização da válvula de *bypass* mantendo a tensão– Duração, 3 minutos;
8. Máquina desligada;

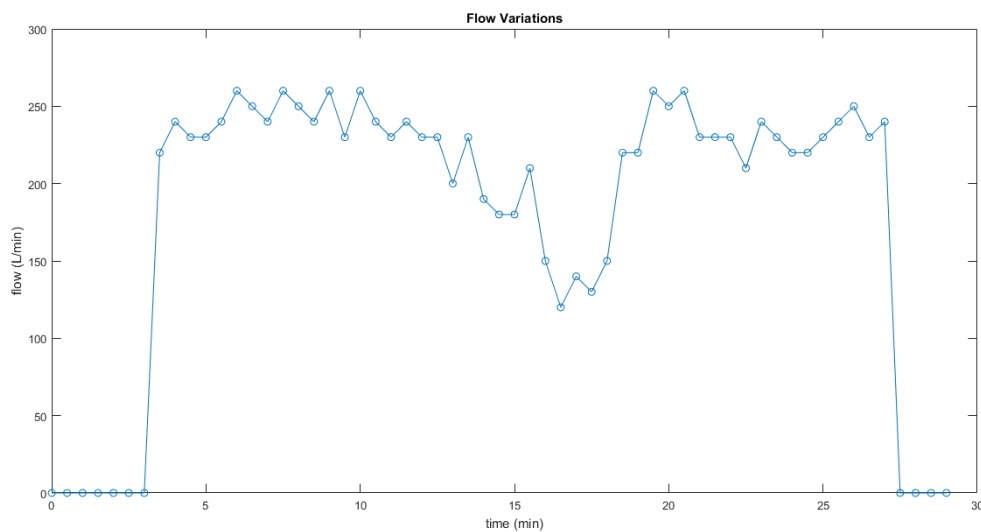


Figura 4.3- Medição do fluxo em modo bypass

Partindo do processo de medição descrito anteriormente e analisando o gráfico da Figura 4.3, é possível encontrar traços correspondentes com as variações efetuadas. Os valores obtidos foram então analisados, registrando os máximos e mínimos sendo feita a média ponderada em cada passo (Tabela 4.1).

Em primeiro lugar serão analisados os valores nos quais a tensão aplicada se encontra no ponto central (passos 2, 5, 6 e 7). Verifica-se que os valores médios encontrados se encontram próximos em condições ditas normais (válvula 2 aberta na totalidade) sendo que, como esperado, ao forçar o encerramento de parte dessa válvula, o caudal disponível diminui, sendo o passo 6 aquele para o qual o valor médio para o ponto central se encontra mais baixo.

Tabela 4.1- Registo das medições de fluxo (L/min) efetuadas

Passo	Maior Valor	Menor Valor	Média Ponderada
1	0	-	0
2	260	220	241,05
3	230	180	198
4	150	120	138
5	260	220	237,5
6	240	210	224
7	250	230	238
8	0	-	0

Na Figura 4.4 encontram-se expostas as variações das diferentes temperaturas, obtidas durante o período de medição.

Enquanto a temperatura ambiente se mantém praticamente constante é possível verificar um aumento de cerca de 3 °C na temperatura da água. O aumento era esperado tendo em conta o contínuo movimento da água no sistema.

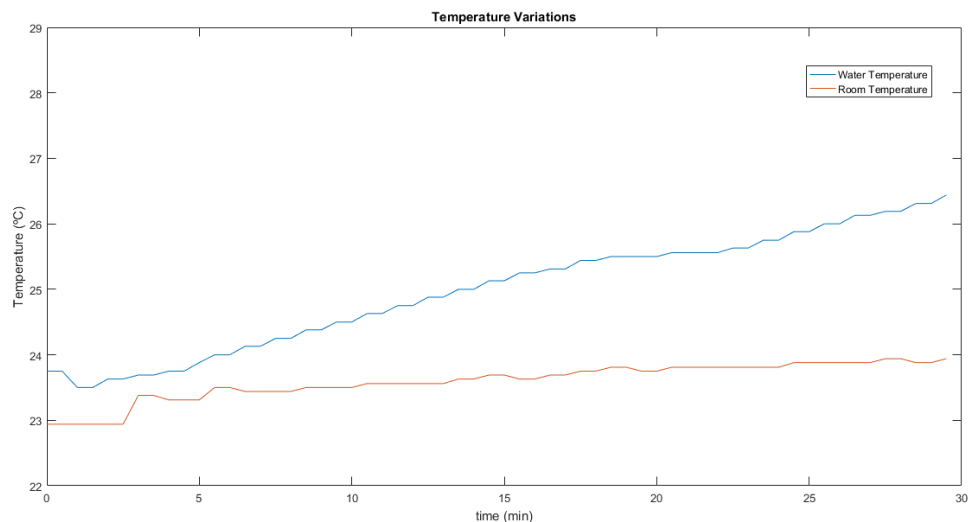


Figura 4.4- Medição das temperaturas em modo bypass

O gráfico da Figura 4.5 claramente demonstra as diferenças de pressão entre um ponto do sistema onde existe fluxo de água (sensor de entrada) e outro onde não existe (sensor de saída). Podemos observar, em ambos os casos, uma clara convergência a partir do minuto 15. O sensor de saída, não estando afetado pelo fluxo de água, obtém valores próximos da pressão atmosférica ($\cong 100\text{ kPa}$) enquanto o sensor de entrada tende para os valores de pressão de $\cong 106\text{ kPa}$.

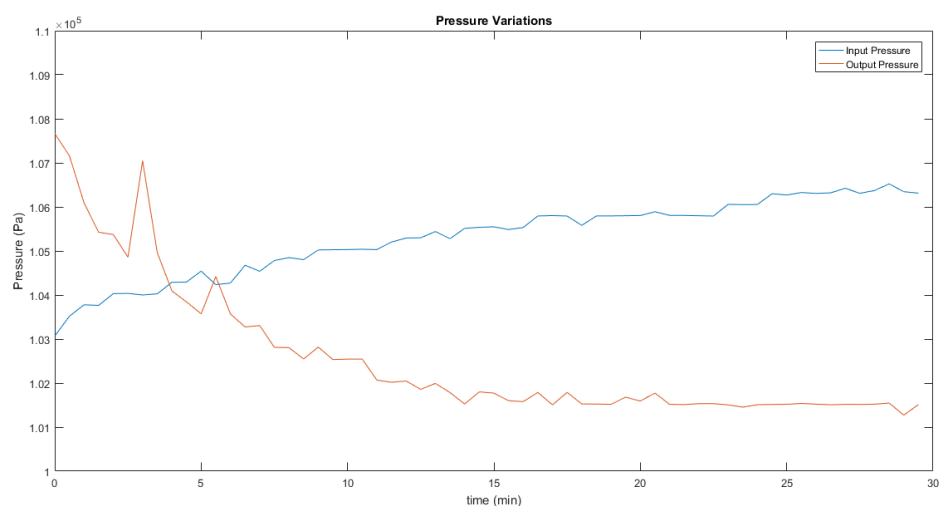


Figura 4.5- Variação das pressões em modo bypass

4.2. PaT

Primeiramente, de maneira a permitir a passagem de água unicamente pela PaT, foi fechada a válvula 2 abrindo as válvulas 3 e 4.

Após a confirmação no capítulo anterior (4.1.) da existência de uma resposta eficaz do sensor de caudal às variações efetuadas, é espectável, neste capítulo, a demonstração da validade de medições por parte dos sensores de pressão.

A bomba PaT, não tendo, nos seus terminais nenhuma fonte de alimentação ativa, funcionará como um elemento resistivo onde é espectável que se verifique uma diferença de pressão consistente com os manómetros analógicos previamente existentes no sistema.

A medição das pressões no modo PaT apresentada na Figura 4.6 seguiu os seguintes passos:

1. Máquina desligada;
2. “PaT simples” (válvulas 3 e 4 abertas, 2 fechada) - Duração, 5 minutos;
3. PaT com *bypass* (válvulas 2, 3 e 4 abertas) - Duração, 4 minutos;
4. Fechar válvula 4- Duração, 5 minutos;
5. PaT com *bypass*- Duração, 5 minutos;
6. Redução da válvula de *bypass* (válvula 2) - Duração, 3 minutos;
7. PaT com *bypass*- Duração, 3 minutos;
8. Máquina desligada.

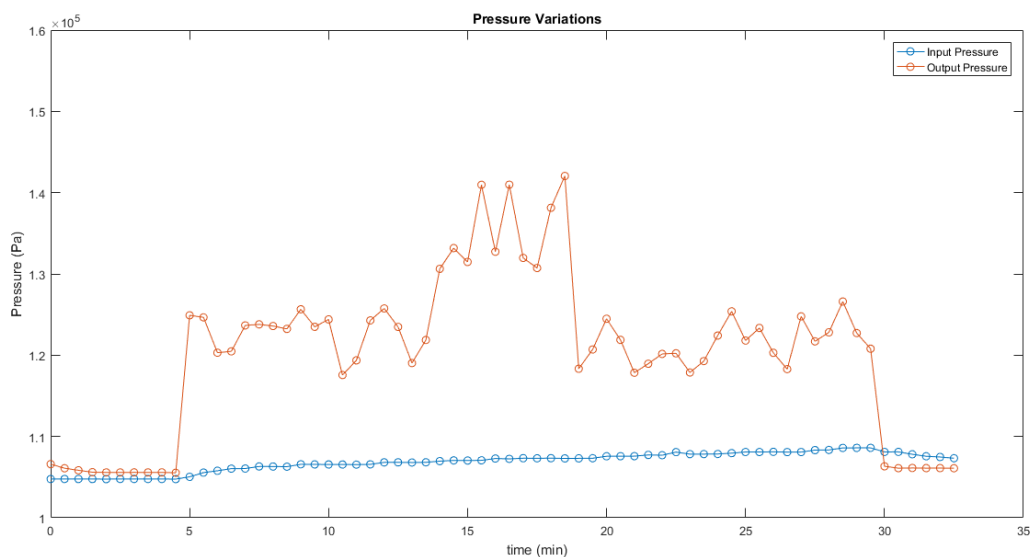


Figura 4.6- Variação das pressões em modo PaT

Como é possível verificar na Figura 4.6, existe uma discrepância notável entre as características de variação obtidas pelo sensor de saída e pelo sensor de entrada.

Observando a característica do sensor de saída, podemos comprovar que esta segue, em média, o padrão de variações implementado. Contudo, onde era esperado a existência de uma redução de pressão entre os pontos de entrada e saída da PaT (nos pontos de “PaT simples”), verifica-se que os valores de pressão para a entrada se mantêm sempre abaixo dos de saída (com exceção aos pontos 1 e 8 onde ambos apresentam valores próximos da pressão atmosférica).

Tabela 4.2-Registro das medições de pressão (kPa) efetuadas

Passo	Pressão Entrada			Pressão Saída			ΔP ($\overline{P_{OUT}} - \overline{P_{IN}}$)
	Maior	Menor	Média	Maior	Menor	Média	
1	104,8	104,8	104,8	106,6	105,5	105,76	0,96
2	106,6	105	106,06	125,6	120,3	123,38	17,32
3	106,8	106,5	106,68	125,8	117,6	121,6	14,92
4	107,3	107	107,2	142,1	130,6	135,29	28,09
5	108,1	107,3	107,62	124,5	117,8	119,97	12,35
6	108,1	107,8	108,03	125,4	118,3	121,9	13,87
7	108,6	108,1	108,41	126,6	120,8	123,2	14,79
8	108,1	107,3	107,73	106,3	106,1	106,13	-1,6

Uma análise mais pormenorizada aos valores resultantes da experiência (Tabela 4.2) confirma as conclusões acima efetuadas.

Os valores obtidos pelo sensor de saída permanecem praticamente constantes durante os pontos 2 e 3 sendo que neste último (a partir do minuto 10) existem algumas reduções de pressão fruto da abertura da válvula 2. Esta ação provoca uma divisão do caudal no sistema podendo causar reduções de pressão no sensor de saída. A variação mais visível ocorre, como esperado, quando é fechada a válvula 4, depois do sensor de saída. A pressão acumulada faz disparar o valor obtido sendo que após a reabertura da válvula (minuto 18) ocorre uma normalização dos valores.

No caso do sensor de entrada, é possível verificar que a sua resposta às variações impostas no sistema é subtil ou, em alguns casos, inexistente. Tal comportamento demonstra a existência, visto que a transmissão e processamento é feita de igual modo para os dois sensores, de um problema na obtenção do valor. Devido à natureza da montagem do sensor no sistema (Figura 3.16), é possível perceber algumas das razões para o qual este problema possa existir:

- Perfuração na tubagem de tamanho insuficiente;
- Desalinhamento entre o sensor e a perfuração;
- Barreira física (plástico, bolha de ar) capaz de causar efeito de atenuação na medição;
- Problema eletrónico no sensor.

Não sendo possível comprovar a redução de pressão devido à falha existente no sensor de entrada, recorreu-se aos manómetros previamente instalados no sistema para confirmar. De maneira a facilitar a visualização do valor nos manómetros foi escolhido o ponto da experiência em que existe maior pressão (ponto 4).

Como é possível verificar na Figura 4.7, no ponto 4, existe uma pressão de 1 bar acima da pressão atmosférica, ou seja, cerca de 200 kPa. À saída, fruto da redução feita pela PaT, existe uma pressão cerca de 0,4 vezes superior à pressão atmosférica (140 kPa). O valor presente no manómetro de saída encontra-se consistente com os valores apresentados pelo sensor de saída no ponto 4.

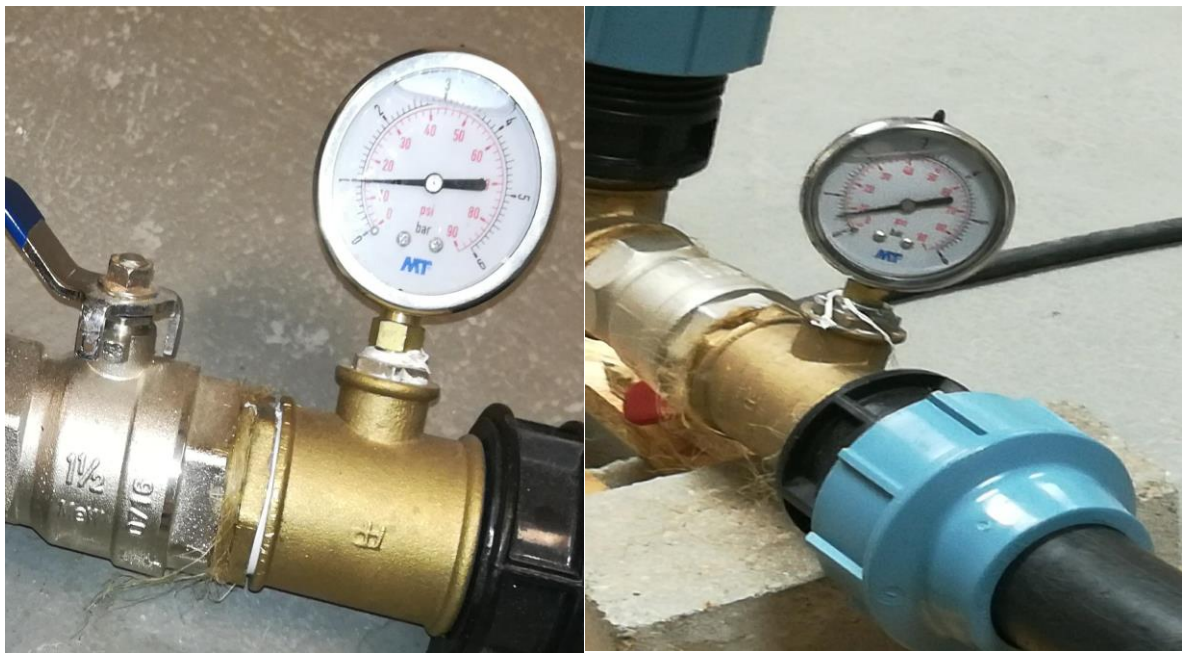


Figura 4.7- Manómetro de entrada (esquerda) e saída (direita) no ponto 4

Conclusões e Trabalhos Futuros

Este capítulo apresenta, em síntese, as necessidades encontradas e o trabalho desenvolvido bem como as limitações que daí provieram.

5.1. Síntese

A adaptação de tecnologias existentes a condições fora do seu âmbito inicial, pode apresentar desafios em termos de definição, conceção e, por consequência, implementação do projeto.

A utilização de bombas de água no seu funcionamento inverso é um exemplo de uma adaptação capaz de solucionar a carência de turbinas aplicáveis a sistemas de baixa produção hídrica. Contudo, a falta de informação sobre o funcionamento de equipamentos para funções para o qual não foram projetados, constitui um problema de dimensionamento bem como de controlo em tempo real.

Nesta dissertação foram desenvolvidas as bases de uma implementação flexível e escalável para um sistema IoT de baixo custo capaz de monitorizar sistemas hídricos PaT. O sistema foi dividido em três grupos: obtenção e processamento de dados, comunicação e armazenamento. As variáveis obtidas, escolhidas devido à preponderância quer no funcionamento do sistema como na segurança do mesmo, foram a pressão, o caudal e a temperatura.

Os sensores escolhidos adequaram-se à especificidade do sistema e, com exceção do problema identificado e imputado à sua montagem, forneceram informações fidedignas sobre o funcionamento do sistema em tempo real.

O nó de monitorização possui, na fase de armazenamento, uma arquitetura de simples compreensão adequada ao sistema a monitorizar, sendo que, não só reúne os dados provenientes dos sensores como também proporciona a primeira fase de alerta em caso de problemas no sistema.

Em suma, o nó de monitorização implementado fornece, salvo uma melhoria na implementação física de um sensor, informações fidedignas com capacidade de se tornarem uma ferramenta útil no controlo e atuação neste e noutros sistemas semelhantes.

5.2. Trabalhos Futuros

De maneira a tornar o sistema completamente funcional e *stand-alone* é necessário a criação de um *access point* junto ao sistema e a transferência do *software* do servidor para um computador capaz de realizar uma monitorização constante.

Posteriormente, sendo este um sistema de aproveitamento energético, deve ser estudada a implementação de um conversor capaz de alimentar o nó de monitorização, reduzindo assim a necessidade de ligação à rede.

A monitorização desenvolvida poderá ainda representar um passo base na implementação dum mecanismo de controlo automatizado. Este controlo será capaz de, com base nos dados retirados do nó de monitorização, adaptar as fontes do sistema (alimentação das bombas de água) ou mesmo o posicionamento de determinadas válvulas. Estes mecanismos aumentarão a eficiência do sistema posicionando-o, em função de valores obtidos em tempo real, no seu BEP.

Bibliografia

- [1] "Hydroelectric Power Generation." [Online]. Available: https://www.mpoweruk.com/hydro_power.htm#top. [Accessed: 09-Aug-2018].
- [2] S. V. Jain and R. N. Patel, "Investigations on pump running in turbine mode: A review of the state-of-the-art", *Renew. Sustain. Energy Rev.*, vol. 30, pp. 841–868, Feb. 2014.
- [3] S. Derakhshan and A. Nourbakhsh, "Experimental study of characteristic curves of centrifugal pumps working as turbines in different specific speeds." [Online]. Available: www.sciencedirect.com. [Accessed: 08-Aug-2018].
- [4] A. K. Himanshu Nautiyal *, Varun, "Reverse running pumps analytical, experimental and computational study: A review.", *Renew. Sustain. Energy Rev.*, vol. 14, pp. 2059–2067, 2014.
- [5] S. Derakhshan and A. Nourbakhsh, "Theoretical, numerical and experimental investigation of centrifugal pumps in reverse operation." [Online]. Available: http://www.iust.ac.ir/files/mech/derakhshan_5df40/files/ISI-Derakhshan2.pdf. [Accessed: 16-Aug-2018].
- [6] "O que é Arduino? - FilipeFloP." [Online]. Available: <https://www.filpefloP.com/blog/o-que-e-arduino/>. [Accessed: 22-Feb-2018].
- [7] "Buy Raspberry Pi 3 Model B from £28.79 – Compare Prices on idealo.co.uk." [Online]. Available: <https://www.idealco.co.uk/compare/4967815/raspberry-pi-3-model-b.html#gallery>. [Accessed: 22-Feb-2018].
- [8] P. Nagvajara, "AC 2011-2307: DESIGN PROJECTS FOR PROGRAMMABLE EMBEDDED SYSTEM-ON-CHIP COURSE Design Projects for Programmable Embedded System-On-Chip Course." [Online]. Available: <https://pdfs.semanticscholar.org/9b62/7dab3aee5da44c9c6bc290af94104a32196.pdf>. [Accessed: 10-Jan-2018].
- [9] E. M. Vicente, P. Dos Santos, C. A. Da Costa, N. V. T. Bernardes, R. L. Moreno, and E. R. Ribeiro, "DESENVOLVIMENTO DE UM SISTEMA MICROCONTROLADO DE SINAIS MISTOS PARA A SUBSTITUIÇÃO DO TCA785," 2012. [Online]. Available: <http://www.eletrica.ufpr.br/anais/cba/2012/Artigos/100160.pdf>. [Accessed: 10-Jan-2018].
- [10] "CY8CKIT-059 | PSoC 5LP Prototyping Kit | Cypress Semiconductor." [Online]. Available: <https://uk.rs-online.com/web/p/processor-microcontroller-development-kits/1244192/>. [Accessed: 22-Feb-2018].
- [11] R. DA Taveira Gonçalves Silva and C. Gabriel Santos Oliveira, "SISTEMA DE AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO UTILIZANDO O ESP8266." [Online]. Available: http://prpi.ifce.edu.br/nl/_lib/file/doc1243-Trabalho/PEVPI_RF.pdf. [Accessed: 26-Jan-2018].
- [12] "ESP-12 Lua Nodemcu WIFI Dev Board Internet Of Things with ESP8266 | eBay."

- [Online]. Available: <https://www.ebay.in/itm/ESP-12-Lua-Nodemcu-WIFI-Dev-Board-Internet-Of-Things-with-ESP8266-/172695701832?hash=item2835774948>. [Accessed: 22-Feb-2018].
- [13] "Bi-Directional Logic Converter or bridge power | 14core.com." [Online]. Available: <https://www.14core.com/how-to-use-bi-directional-bus-logic-converter-bridge-communication/>. [Accessed: 22-Feb-2018].
- [14] R. H. Walden, "Analog-to-Digital Converter Survey and Analysis," *IEEE J. Sel. AREAS Commun.*, vol. 17, no. 4, 1999.
- [15] C.-C. Liu, S.-J. Chang, G.-Y. Huang, and Y.-Z. Lin, "A 10-bit 50-MS/s SAR ADC With a Monotonic Capacitor Switching Procedure," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 731–740, Apr. 2010.
- [16] "Demystifying Delta-Sigma ADCs," 2016. [Online]. Available: <https://www.maximintegrated.com/en/app-notes/index.mvp/id/1870>. [Accessed: 15-Jan-2018].
- [17] P. A. P. T. Ms.Neha R. Laddha, "A Review on Serial Communication by UART," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 1, pp. 366–369, 2013.
- [18] F. Leens, "An introduction to I2C and SPI protocols," *IEEE Instrum. Meas. Mag.*, vol. 12, no. 1, pp. 8–13, 2009.
- [19] "I2C Bus | USB-I2C/SPI/GPIO Interface Adapters." [Online]. Available: <http://dlnware.com/i2c>. [Accessed: 22-Feb-2018].
- [20] J. Haartsen, M. Naghshineh, J. Inouye, O. J. Joeressen, and W. Allen, "Bluetooth," *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 2, no. 4, pp. 38–45, 1998.
- [21] "Bluetooth tutorial | Bluetooth basics | tutorials." [Online]. Available: http://www.rfwireless-world.com/Tutorials/Bluetooth_tutorial.html. [Accessed: 22-Feb-2018].
- [22] "Bluetooth Core Specification 5.0 FAQ Key features and benefits of using Bluetooth 5.0 Increased bandwidth for Bluetooth® technology with low energy," 2016. [Online]. Available: www.bluetooth.com/brand-guide. [Accessed: 23-Jan-2018].
- [23] Tanenbaum and Wetherall, "Computer Networks." [Online]. Available: [http://www.uoitc.edu.iq/images/documents/informatics-institute/exam_materials/Computer Networks - A Tanenbaum - 5th edition.pdf](http://www.uoitc.edu.iq/images/documents/informatics-institute/exam_materials/Computer_Networks_-_A_Tanenbaum_-_5th_edition.pdf). [Accessed: 21-Jan-2018].
- [24] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards," 2006. [Online]. Available: https://ac.els-cdn.com/S0140366406004749/1-s2.0-S0140366406004749-main.pdf?_tid=fb42c0cc-ff69-11e7-9c82-00000aacb35f&acdnat=1516621811_f38e8885d852ec171e905aae30240f5a. [Accessed: 22-Jan-2018].
- [25] "P2PU | A Hands-on Experience with Wireless Sensor Networks (#handsonwsn) | 3. Introduction to ZigBee." [Online]. Available: <https://courses.p2pu.org/en/courses/831/content/1739/>. [Accessed: 22-Feb-2018].
- [26] "ZigBee Specification," 2005. [Online]. Available:

- <https://www3.nd.edu/~mhaenggi/ee67011/zigbee.pdf>. [Accessed: 12-Feb-2018].
- [27] "Liquid Flow Transmitters With Visual Indication." [Online]. Available: <https://www.omega.com/pptst/FPR200.html>. [Accessed: 22-Feb-2018].
 - [28] "Water Meters for Totalization and Rate Indication." [Online]. Available: <https://www.omega.com/pptst/FTB8000B.html>. [Accessed: 22-Feb-2018].
 - [29] "Flow Measurement Assignment Help - Flow Measurement Homework Help Online." [Online]. Available: <https://www.transtutors.com/homework-help/electrical-engineering/electrical-and-electronic-instrumentation/flow-measurement.aspx>. [Accessed: 22-Feb-2018].
 - [30] "Metric Pressure Transducer." [Online]. Available: https://www.omega.com/pptst/PXM209_PXM219.html. [Accessed: 22-Feb-2018].
 - [31] "Gravity: Analog Water Pressure Sensor - DFRobot." [Online]. Available: <https://www.dfrobot.com/product-1675.html>. [Accessed: 22-Feb-2018].
 - [32] "Temperature Sensor - Waterproof (DS18B20) - SEN-11050 - SparkFun Electronics." [Online]. Available: <https://www.sparkfun.com/products/11050>. [Accessed: 22-Feb-2018].
 - [33] "Monnit Wireless Water Temperature Sensor - Commercial Coin Cell Powered | MNS-9-W1-TS-WT-L03." [Online]. Available: <https://www.monnit.com/Product/MNS-9-W1-TS-WT-L03>. [Accessed: 22-Feb-2018].
 - [34] "DS18x8 Digital Temperature Sensor General description Input-output connections," *PSoC Creat. Compon. datasheet*, pp. 1-9.
 - [35] "Gravity: Water Pressure Sensor SKU: SEN0257." [Online]. Available: [https://media.digikey.com/pdf/Data Sheets/DFRobot PDFs/SEN0257_Web.pdf](https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SEN0257_Web.pdf). [Accessed: 21-Aug-2018].
 - [36] "G2 "DN50 Plástico 2 polegada sensor de fluxo Salão sensor de medidor de Fluxo da turbina da água em Sensores de Fluxo de Ferramentas no AliExpress.com | Alibaba Group." [Online]. Available: https://pt.aliexpress.com/item/G2-DN50-flow-rate-10-350L-min-water-Plastic-Hall-flow-sensor-meter-control-water-liquid/32424336992.html?spm=a2g03.search0104.3.8.1f117f42100uGh&ws_ab_test=searchweb0_0,searchweb201602_5_10065_10068_5015215_10059_10696_10003. [Accessed: 08-Sep-2018].
 - [37] "Bi-Directional Logic Level Converter Hookup Guide." [Online]. Available: [https://media.digikey.com/pdf/Data Sheets/Sparkfun PDFs/Bi-Directional-Logic-Level_HookupGuide.pdf](https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/Bi-Directional-Logic-Level_HookupGuide.pdf). [Accessed: 20-Aug-2018].
 - [38] "Low-cost embedded development kits accelerate design cycles." [Online]. Available: <http://www.embedded-computing.com/embedded-computing-design/low-cost-embedded-development-kits-accelerate-design-cycles>. [Accessed: 02-Jul-2018].
 - [39] "Adafruit Feather HUZZAH with ESP8266 WiFi - Micro Robotics." [Online]. Available: <https://www.robotics.org.za/AF2821>. [Accessed: 05-Jul-2018].



Código PSoC

```
#include "project.h"
#include <stdio.h>

void Compare_Interrupt(void)
{
    // Call the ReadStatusRegister to clear the Interrupt Status bit //
    Window_ReadStatusRegister();
    // Set the flag to indicate occurrence of the interrupt //
    int compare_occured = 1;
    // `#END` //
}

int main(void)
{
    CyGlobalIntEnable;
    char send_rt[10], send_wt[10], data[100];
    int32 pressureIN, pressureOUT;
    char* room_temperature;
    char* water_temperature;
    uint16_t freq;
    float flow;
    int flag_timer=1;

    Opamp_1_Start();
    ADC_Pressure_Start();
    Mux_Start();
    UART_Start();
    DS18B20_reader_Start();
    Counter_Start();
}
```

```

Window_Start();
Compare_Interrupt_Start();

for(;;)
{
    ////////////PRESSURE//////////
    Mux_FastSelect(0); //IN Pressure
    ADC_Pressure_StartConvert();
    ADC_Pressure_IsEndConversion(ADC_Pressure_WAIT_FOR_RESULT);
    pressureIN=ADC_Pressure_GetResult32(); //Pa
    ADC_Pressure_StopConvert();

    Mux_FastSelect(1); //OUT Pressure
    ADC_Pressure_StartConvert();
    ADC_Pressure_IsEndConversion(ADC_Pressure_WAIT_FOR_RESULT);
    pressureOUT=ADC_Pressure_GetResult32(); //Pa
    ADC_Pressure_StopConvert();
    ////////////

    ////////////Temperature//////////
    if(flag_timer){
        flag_timer=0;
        DS18B20_reader_SendTemperatureRequest();
    }
    if(DS18B20_reader_DataReady){
        DS18B20_reader_ReadTemperature();
        room_temperature= DS18B20_reader_GetTemperatureAsString(0);
        sprintf(send_rt,"%s",room_temperature);
        water_temperature= DS18B20_reader_GetTemperatureAsString(1);
        sprintf(send_wt,"%s",water_temperature);
        flag_timer=1;
    }
    ////////////

    ////////////Flow//////////
    freq= Counter_ReadCapture()*2;
    flow= (float)freq/0.2;
    ////////////

    sprintf(data,{"\pIN\":%.2ld,\pOUT\":%.2ld,\rt\":%s,\wt\":%s,\f\":%.2f? ",pressureIN, pres-
sureOUT,send_rt,send_wt,flow);
    UART_PutString(data);
    CyDelay(30000);
}
}

```



Código ESP8266

```
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
```

```
const char* ssid    = "HUAWEI_P8";
const char* password = "123456789";
```

Definições da rede *Wi-Fi*

```
SoftwareSerial psoc(12,13); //rx,tx
```

Definições por *software* dos pinos de comunicação

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  while(!Serial){
  }
  pinMode(LED_BUILTIN, OUTPUT);
  psoc.begin(115200);
  while(!psoc){
  }
  delay(100);

  psoc.setTimeout(2000);
```

```
WiFi.begin(ssid, password);
```

```
while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
}  
}
```

Verificação da ligação à rede *Wi-Fi*

```
String data="";
```

```
char character;
```

```
bool endofmessage = false;
```

```
String dataToSend = "";
```

```
bool messageSent = true;
```

```
int count = 0;
```

```
void loop() {
```

```
digitalWrite(LED_BUILTIN, LOW);
```

```
if(psoc.available()) {  
    character=psoc.read();  
    if(character == '?') {  
        endofmessage = true;  
    } else {  
        endofmessage = false;  
        data.concat(character);  
    }  
}
```

Leitura dos caracteres do *stream* e
concatenação até ser atingido o caracter
terminal ‘?’

```
if(character == '{'){  
    count=count+1;  
}
```

Verificação de integridade da mensagem
(presença de apenas um caracter iniciante)

```
}
```

```
if(endofmessage == true){
```

```
Serial.println(data);
```

```
if (count == 1){
```

```
dataToSend = "";
```

```
dataToSend.concat(data);
```

```
messageSent= false;
```

```

    }
    data = "";
    count = 0;
    endofmessage = false;
}

```

```

if (WiFi.status() == WL_CONNECTED && messageSent == false) {
    digitalWrite(LED_BUILTIN, HIGH);

```

```

    HTTPClient http; //Declare an object of class HTTPClient

```

```

    http.begin("http://192.168.43.24:10010/post");

```

Definição do IP, porto e método do servidor

```

    http.addHeader("Content-Type","application/json");

```

Adição do cabeçalho de conteúdo à mensagem

```

    delay(100);
    int httpCode = http.POST(dataToSend);
    String payload = http.getString();
    delay(100);
    messageSent= true;
    http.end();
}
}

```




Código Servidor

```
/* Código Servidor a ser executado mediante a ativação de um evento prove-  
niente do ESP8266*/
```

```
/*Declaração das bibliotecas utilizadas no código*/
```

```
var util = require('util');  
var models = require('../DataBase/models.js');  
var moment = require('moment');  
var fs = require('fs');  
var csvWriter = require('csv-write-stream');  
var lines=0;
```

```
function post(req, res) {
```

```
/* Obtenção do corpo da mensagem, em formato JSON, da variável recebida  
(req.body) e preenchimento das variáveis respectivas (input)*/
```

```
var esp_input = req.body;  
var limits = [50,50,1600000,1600000, 350]; //room_temp_max, wa-  
ter_temp_max,pre_max_IN(1.6MPa),pre_max_OUT(1.6MPa), 200 L/min  
var input = [esp_input.rt, esp_input.wt, esp_input.pIN, esp_input.pOUT,  
esp_input.f];
```

```
/*Criação da entrada newRecord*/
```

```
var newRecord = new models.Data_Regist({ //Criação de entrada newRecord  
pressure_in: esp_input.pIN,  
pressure_out: esp_input.pOUT,  
room_temp: esp_input.rt,  
water_temp: esp_input.wt,
```

```

    flow: esp_input.f,
    timestamp: moment().locale('pt').format('l') + ' ' + moment().lo-
cale('pt').format('LTS'),
  });

/*Gravação da entrada newRecord na Base de Dados*/

newRecord.save(function(err){
  if(err){
    console.log("Error Saving New Record");
    console.log(err);
  }else{
    console.log("New Record Created!");
  }
});

/*Escrita no ficheiro csv das variáveis de controlo do sistema*/

if (!fs.existsSync('data.csv')){ //início da gravação no ficheiro csv
  var writer = csvWriter({ headers: ["pIN", "pOUT", "rt", "wt", "f"]});
  writer.pipe(fs.createWriteStream('data.csv'));
  writer.write([esp_input.pIN, esp_input.pOUT, esp_input.rt, esp_input.wt,
esp_input.f]);
  writer.end();
  lines++;
}
else{
  if(lines == 120){ //Número de entradas desejadas no csv
    var writer = csvWriter({ headers: ["pIN", "pOUT", "rt", "wt", "f"]});
    writer.pipe(fs.createWriteStream('data.csv'));
    writer.write([esp_input.pIN, esp_input.pOUT, esp_input.rt, esp_in-
put.wt, esp_input.f]);
    writer.end();
    lines=1;
  }else{
    var writer = csvWriter({sendHeaders: false});
    writer.pipe(fs.createWriteStream('data.csv', {flags: 'a'}));
    writer.write({pIN: esp_input.pIN, pOUT: esp_input.pOUT, rt: esp_in-
put.rt, wt: esp_input.wt, f: esp_input.f})
    writer.end()
    lines++;
  }
}
}

```


/*Criação da entrada realTime caso ainda não exista entrada anterior (verificação "findOne"), caso exista procedesse à atualização da entrada existente*/

```
models.Real_Time.findOne({}, function (err, model) { //verificação realTime
  if (model == null) {
    var real_time = new models.Real_Time({ //criação realTime
      pressure_in: esp_input.pIN,
      pressure_out: esp_input.pOUT,
      room_temp: esp_input.rt,
      water_temp: esp_input.wt,
      flow: esp_input.f,
    });

    real_time.save(function(err){
      if(err){
        console.log("Error Saving Real Time Record");
        console.log(err);
      }else{
        console.log("Real Time Record Created!");
      }
    });
  } else { //atualização realTime

    model.pressure_in= esp_input.pIN;
    model.pressure_out= esp_input.pOUT;
    model.room_temp= esp_input.rt;
    model.water_temp= esp_input.wt;
    model.flow= esp_input.f;

    model.save(function(err){
      if(err){
        console.log("Error Updating Real Time");
        console.log(err);
      }else{
        console.log("Model Updated!");
      }
    });
  }
});
```

/*Chamada da função allerts com parâmetros os limites máximos definidos e os valores de sistema recebidos*/

```
allerts(limits, input); //chamada função de verificação de alertas
}
```

```

function allerts(limits, input){
    var sensor;

    for(var x=0; x < input.length; x++){
        /*Verificação de bom funcionamento do sensor
        Caso existam falhas no sensor o valor recebido será "null" */

        if(input[x] == null){ //receção de valores nulos
            switch(x){
                case 0:
                    sensor= "Room Temperature Sensor";
                    break;
                case 1:
                    sensor= "Water Temperature Sensor";
                    break;
                case 2:
                    sensor= "Input Presure Sensor";
                    break;
                case 3:
                    sensor= "Output Pressure Sensor";
                    break;
                case 4:
                    sensor= "Flow Sensor";
                    break;
            }
        }
        /*Criação da entrada alert*/

        var alert = new models.Alert({ //criação de alerta
            parameter: sensor,
            description: "Sensor Malfunctioning",
            value: "No Value",
            timestamp: moment().locale('pt').format('l') + ' ' + moment().lo-
cale('pt').format('LTS'),
        });

        /*Gravação da entrada alert na Base de Dados*/

        alert.save(function(err){
            if(err){
                console.log("Error Saving Alert");
                console.log(err);
            }else{
                console.log("Alert Created!");
            }
        }, 200);
    }
}

```

```

/*Verificação dos limites do sistema
Caso os valores recebidos ultrapassem os limites, é criado e guardado um
alerta */

if(input[x]>limits[x]){ //verificação se valores se encontram nos limites
  switch(x){
    case 0:
      sensor= "Room Temperature";
      break;
    case 1:
      sensor= "Water Temperature";
      break;
    case 2:
      sensor= "Input Presure";
      break;
    case 3:
      sensor= "Output Pressure";
      break;
    case 4:
      sensor= "Flow";
      break;
  }

/*Criação da entrada alert*/

  var alert = new models.Alert({
    parameter: sensor + " sensor",
    description: "High" + sensor+ " values",
    value: input[x],
    timestamp: moment().locale('pt').format('l') + ' ' + moment().lo-
cale('pt').format('LTS'),
  });

/*Gravação da entrada alert na Base de Dados*/

  alert.save(function(err){
    if(err){
      console.log("Error Saving Alert");
      console.log(err);
    }else{
      console.log("Alert Created!");
    }
  }, 200);
}
}

```